MULROONEY, TIMOTHY J., Ph.D. Turning Data into Information: Assessing and Reporting GIS Metadata Integrity Using Integrated Computing Technologies. (2009) Directed by Dr. Rick Bunch. 250 pp.

A Geographic Information System (GIS) serves as the tangible and intangible means by which spatially related phenomena can be created, analyzed and rendered. GIS metadata serves as the formal framework to catalog information about a GIS data set. Metadata is independent of the encoded spatial and attribute information. GIS metadata is a subset of electronic metadata which catalogs electronic resources such as web pages and software applications. However, GIS metadata is inherently different than electronic media because each metadata file can be applied to a spatial component that is not implicit with other forms of metadata.

Using open source technologies such as R, Perl and PHP, metadata information for large GIS data sets (thousands of layers) can be gleaned quickly and more efficiently than the human element. In doing so, metrics to express the integrity of both the metadata and GIS data can be captured, displayed and compared for use in the decision making process. Supervised and unsupervised techniques allow users and computer algorithms to explore unseen trends about the GIS data not obvious to the human component.

The validity of these analyses was tested using a Technology Acceptance Model (TAM). Responses from 40 GIS professionals about the results of this methodology were captured to find a relationship between this technology's Perceived Ease of Use, Perceived Usefulness, Attitude Towards Using and the Intention to Further use this technology.

TURNING DATA INTO INFORMATION: ASSESSING AND REPORTING GIS METADATA INTEGRITY USING INTEGRATED COMPUTING TECHNOLOGIES

by

Timothy J. Mulrooney

A Dissertation Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Greensboro
2009

Approved by

_____
Committee Chair

APPROVAL PAGE

This dissertation has been approved by the following committee of the Faculty of The Graduate School at the University of North Carolina at Greensboro.

Committee Chair _____

Committee Members _____

_____

_____

_____

_____
Date of Acceptance by Committee

_____
Date of Final Oral Examination

ACKNOWLEDGMENTS

This dissertation represents the culmination of years of hard work both inside and outside of the classroom. I have many to thank who have aided in both my formal and informal education in geography, computer science and the problem solving skills it takes to solve a research problem of this magnitude.

I would like to thank my dissertation committee of Dr. Rick Bunch, Dr. Roy Stine, Dr. ZJ Liu and Dr. Hamid Nemati. Dr. Bunch took valuable time from his already busy schedule to mentor me. When I was available to work over the summer, he availed himself and worked vigorously to arrange comprehensive exams and my oral defense with other committee members. I am extremely grateful for his efforts, as they extended above and beyond those expected for one in his position.

On the home front, I would like to thank my father, Dr. Tim Mulrooney for his help and guidance in these subjects. Through his work as an electrical engineer, he has been exposed to many of the open source programming languages. While he is not familiar with the geospatial component of this research, he is well versed in the many digital environments in which this complex problem can be solved. Also on the home front, I would like to thank my mother Celeste for her support.

Lastly, I would like to recognize my brother Shawn, who was a graduate student in the Geography Department at UNCG until his passing in March of 2009. When I ran into some personal roadblocks while living in Virginia, he suggested that I move to Greensboro so I could complete my PhD studies. While he joked that I would probably not be finished until around 2012, I am sure that he would be proud.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

A Geographic Information System (GIS) serves as the tangible and intangible means by which information about spatially related phenomena can be stored, analyzed,and mapped. Experts in many dissimilar fields have seen the utility of GIS as a means of quantifying and expanding their research. GIS is used in disciplines such as business, sociology, justice studies, surveying and the environmental sciences (Steinberg and Steinberg 2006). In fact, most data can have a spatial component. The manner, however, in which we capture this spatial element varies. Some methods include using a GPS (Global Positioning System) unit, extracting or improving existing GIS data, or creating data from an analog format using a process called digitization. Regardless of the method, the resources (e.g., the computers, time and people dedicated to the process of collecting and creating GIS data) are the most time consuming portion of a GIS-related project. As a result, the GIS community needs to ensure that the quality of the GIS data created as a result of these methods is captured and assessed in a systematic way. Programmers can employ programming technologies and evolving data mining techniques not previously used within GIS metadata as a means to accomplish this.
The proliferation of geospatial sciences throughout the world has increased the number of companies developing GIS software. Thus, there currently is no all-encompassing data standard or format employed by the entire GIS user community. One of the results

these developments is a compatibility issue with other companies' proprietary data formats. The area of research that addresses this is GIS interoperability and it is recognized as an area ripe for further research because it examines information sharing possibilities and its potential contribution to the degradation of GIS data quality (Reichardt 2005), both of which must be catalogued using some mechanism. Metadata is one such mechanism, and it exists to document the integrity of these routines and data for approval by the larger GIS user community. In addition to the artificial processes associated with GIS data, information about the data creators, date of creation, data steward, data format and updates performed on the data set also do not fit within the framework of tabular attribution or interoperability non-repudiation files, but still must be retained. Furthermore, processing steps need to be appended and contacts updated when additional processing is performed on the data or personnel changes occur. This qualitative and quantitative information is also stored in metadata.

Metadata, most generally thought of as "data about data", serves as a formal framework to catalog the lifeline of a particular GIS data set. It is independent of the encoded spatial and attribute information, and depending upon the GIS software and data format, may be stored in an independent field in a RDBMS (Relational Database Management System) or an entirely separate file. Metadata is usually thought of as a necessary evil in the world of GIS (Leiden, et al. 2001), but some feel that GIS data alone without accompanying metadata is worthless (Qi, et al. 2004). Like other technology-based industries, GIS is prone to experience high personnel turnover. As a result, metadata serves as a record so a seamless transition can be made from one data steward

to the next. Without it, Qi and colleagues (2004) are correct; the quality of a GIS product is proven, in part, by the documented data with which it is made.

Metadata is populated when a data set is processed or updated. Consider the following three examples. If well locations are collected using a GPS unit, all parameters regarding the GPS unit suh as environmental conditions, personnel collecting the data and date of collection should be retained. If lakes are digitized from ortho imagery, information about the source (date of collection, agency collected from, etc) must be retained in addition to the processes used to create the digital data set. If a roads layer is updated with new road names, the appropriate metadata elements (processing step, process contact, data currentness) should also be updated. Maintaining a complete and comprehensive metadata database is a continual and interactive process. When data need to be mapped or analyzed, then, it is important that all personnel working on the GIS data have a means to assess the spatial, attribute and temporal integrity of the data.

Like all facets of GIS, advances in computing technology have extended into the realm of metadata. Metadata population is much easier and accurate than it was with previous generations of GIS software. Technologies now exist with the sole purpose of creating, editing and exporting GIS metadata. Projection and attribute information, for instance, can be gleaned from the spatial information and auto-populated into the metadata. However, information germane to a particular GIS data set, such as creation date, process steps and contact information are proprietary to the individual data set and must be populated as such. The FGDC (Federal Geographic Data Committee) regularly meets to determine all possible values, parameters and domains that can be captured and

expressed within the confines of GIS metadata. The FGDC serves as a governing body for geospatial data and metadata in the United States. The FGDC defines metadata as the following:

> A metadata record is a file of information, usually presented as an XML document, which captures the basic characteristics of a data or information resource. It represents the who, what, when, where, why and how of the resource. Geospatial metadata are used to document geographic digital resources such as Geographic Information System (GIS) files, geospatial databases, and earth imagery. A geospatial metadata record includes core library catalog elements such as Title, Abstract, and Publication Data; geographic elements such as Geographic Extent and Projection Information; and database elements such as Attribute Label Definitions and Attribute Domain Values. (FGDC 2000)

Facets of FGDC paradigms and doctrine have been accepted and implemented by other governing agencies such as ISO (International Standards Organization) and even other countries. Failure to populate these metadata elements fully and dutifully in a timely manner may compromise the integrity of projects that implement these data (Qi, et al. 2004). Any conscientious GIS business model should budget adequate time and personnel resources to the creation and update of its metadata.

FGDC metadata standards dictate that a plethora of entries are populated for compliant GIS metadata (FGDC 2000). Thus, ensuring metadata integrity for large spatial data sets is a time-consuming process if done by hand. It is not uncommon for GIS shops or companies to employ thousands of individual data layers within their digital warehouses. Each of these data layers requires a separate metadata file. Sifting through multiple levels of this information has proven to be a time consuming and error prone task for a human (Leiden et al. 2001). In addition, business plans may dictate revisit

times for particular data layers. For example, a roads data layer may need to be updated every year, while census data may need to be updated only every 10 years. How does one analyze the latest processing steps to ensure that the data fit within this paradigm? How can large-scale analysis be performed on these metadata in a timely and efficient manner? Does this information portray trends or tendencies not obvious to the human observer?

Lastly, given the increasingly large size of these data sets, it is difficult if not impossible to make associations between various fields such as publisher, publication date, and theme. In conjuction with this research, data mining techniques made associations or rules between these different fields so sound business decisions can be made. The ultimate goal is to turn GIS metadata into action. The research addressed that goal by exploring a methodology to perform large-scale metadata analysis that would take days or weeks to do by a human. It also defined metrics to describe the current state of metadata and its relationship to defined standards so prudent business decisions can be made in the future. The increasing schism between the rate at which data are created and the efficiency at which the metadata are assessed was the cornerstone of this research. There currently exist very few technological constructs to assess metadata integrity for large data sets in all or even most software environments. While Greenberg (2006) and Craven (2001) admit that human interaction in this process is required, it must be integrated with a technological piece. Current metadata QA/QC (Quality Assurance/Quality Control) procedures rely heavily on the human component to verify the integrity of GIS metadata. This work extended the concepts of digital data mining

and large-scale data analysis so that this human component serves more as a capable interpreter to the information discovered from this analysis. In doing so, the following topics were examined:

- The pressing need for GIS metadata and the appropriate government standards applied to GIS metadata
- The relationship between GIS, metadata and statistical programming technologies within the digital environment which are independent of software and operating system environments
- The statistical metrics that can best be applied to GIS metadata
- The extension of the concepts of large-scale data analysis and data mining to the realm of GIS metadata to discern apparent and hidden trends within large GIS metadata sets
- The creation of an XML standard so results from this analysis can be shared across the entire GIS user community
- The interpretation of these results that can be applied to the business process
- The dissemination of these results under one cohesive umbrella

The collusion of statistical and web technologies within the realm of GIS metadata was able to calculate and disseminate information, insight and trends about large data sets. Because of the high costs associated with its creation, the ultimate goal of any GIS data layer should be integration and sustainability. While integration speaks to the very nature of the data layer, sustainability can not be achieved if the metadata are inadequately populated. This research created a means to evaluate the effectiveness and

efficiency of these methods over traditional methods. Any GIS project is only as good as the information on which it is based, while some go so far to say that GIS data and therefore the proceeding analysis is worthless without the accompanying metadata (Qi et al. 2004). This research explored methodology and techniques to ensure this sustainability through standards compliance, analysis of applicable metrics and the discernment of trends using digital media and techniques not previously seen for large scale analysis. The analysis of these data, metric definition and extraction of rules and associations served as the intellectual motivation of this research.

The difference between a basic and experienced level GIS software user is his or her ability to integrate parallel technologies to increase computing capabilities. This is sometimes referred to as 'tightly' and 'loosely' coupled processing. In addressing these topics, this research coupled disparate computing technologies such as Perl, statistical programming languages (R) and the web under one unified umbrella so end users can seamlessly assess and evaluate their GIS metadata. An added bonus to using these methods is their lack of reliance on particular software environments and operating systems. These specific approaches served as the technological benefits of this research. As a result, the methodologies discussed here can be repeated within all normal computing environments and replicated for future use, updates and customization.

CHAPTER II

LITERATURE REVIEW

This work converges at the fields of GIS, metadata and statistical programming. It will extend the concepts and theories of data mining and large-scale data analysis to assess and report the integrity of metadata for large GIS data sets. As part of this process, it will bridge the chasm between the rate at which data are created and the dutiful reporting of this information. These fields are evolving as their need and scope dictate. Previously, scientists were limited by processing speeds unable to tackle the size of these data sets. Because of their reliance on digital technology, these fields are relatively new and paradigm shifts occur in concert with major advances in hardware and software technology. The popularity of web-based applications such as Google Earth has proven that the delivery of high volumes of spatial data is possible and feasible. This review explores how predecessors evaluated metadata elements and metadata integrity for large data sets within the digital environment.

Metadata and Statistical Programming

The different components of metadata can be assessed a variety of different ways. The frequency of keywords can be counted among all metadata files. The horizontal accuracy can be averaged and compared among those for other databases Technologies to apply statistical metrics to these components of metadata were the focus of this research. Metadata serves as an organized means to describe a data set, and it provides

the formal framework for reporting information about a data set's lineage, age, and creators. Metadata are composed of both qualitative and quantitative information. In addition to these descriptive features, GIS metadata has the extra job of capturing spatial information. As this relates to geography, metadata helps to encapsulate the five fundamental themes of geography within a formal framework. Metadata has constructs to describe location, place, movement, human-environment relations and its relationship with larger regions. Other forms of metadata described below do not necessarily need to worry about spatiality and the effects of spatiality. GIS metadata must include constructs that these other forms of metadata do not need.

Non-spatial metadata such as *Darwin Core* serves as a means to catalog museum specimens (Haas 2003). Closer to the realm of GIS, *MARC* (Machine Readable Cataloging) and its successor *MARC 21* are used by the Library of Congress to catalog bibliographic resources. This system has been in place since the 1960s, but it was not originally designed for computer interfacing, and the format is not very intuitive (Library of Congress 2008). A more popular format called the *Dublin Core* was created in 1995 for electronic recourses such as web pages and software applications. It was created in 1995, so the FGDC (Federal Geographic Data Committee) and GIS metadata standards described below actually predate this more generalized format (Dublin Core 2008).

Dublin Core and FGDC generally share a base level of descriptive metadata elements. While Dublin Core is used to describe electronic resources, GIS metadata adheres to FGDC requirements. These requirements are always changing as dictated by technology. Because of the spatial nature of GIS data, FGDC requirements dictate that

information pertaining to absolute location be retained. These fields include datum, coordinate system, false easting, false northing and bounding coordinates. While Dublin Core does make accommodations for place keywords and spatial descriptors, it does not contain placeholders for elements that help describe geodetic elements with such detail as GIS metadata.

Because of the different goals of each standard, a preciptious balance between MARC, Dublin Core and FGDC Metadata must be found. Crosswalking, a tedious and sometimes imprecise process where either people or algorithms find matching elements between the different standards may be necessary because various organizations use these popular formats interchangeably on a routine basis. Reese (2005) explored methods to adhere to standards outside of FGDC, such as examining the feasibility of compatibility with the Dublin Core metadata standard. In a related work, Batcheller (2008) attempted to crosswalk between Dublin Core and FGDC metadata elements using the digital means previously described.

MARC is an excellent way to catalog a multitude of information, but now statistical programming techniques and technology allow savvy programmers to sift through this information and aid, if not replace, the decision-making process. While up-front costs, such as application development, are incurred in the automation of metadata retrieval and assessment within a digital library, the general opinion in the metadata community is that it is an effective alternative to the human component. Greenberg (2006) cites that it is ‘‘unrealistic to depend on traditional humanly generated metadata approaches’’ when attempting to assess metadata integrity. However, a happy medium

must be found between quality assurance, quality control and the necessary human component involved in this process that cannot be replicated in the digital environment. While Anderson and Perez-Carballo (2001) subscribe to the mechanization of metadata assessment processes as the most effective and efficient, others such as Schwartz (2002) and Craven (2001) concede that metadata is best managed through the integration of the human and digital components. While the level of human interaction in this process should be minimal, it should not be eliminated altogether.

Current research in the field of metadata is most closely associated with statistics and high speed processing. Given the exponential increase in electronic resources and media, technologies must be able to accommodate the automation of resources that are viewed, accessed, and assessed. Stvilia and Gasser (2008) examined the role of metadata and its ability to be assessed. They argued that metadata for metadata's sake does no good. Metadata must have some utility as it needs to be assessed and have a role within the decision making process. Metadata must ultimately serve a purpose and specifically the greater good of the user community. While researchers such as Bruce and Hillman (2004) do propose a quality assessment for metadata, they fail to do so with regards to changes in metadata quality, their accompanying values and the holistic structure used to store them. This research aims to address this concern by creating an XML standard which stores output from this assessment. These XML files can be compared from one time period to the next. One of these structures is through ontology, a semantic representation of a concept through various domains and properties. Most recently, Lee et al. (2008) applied e-learning technologies to these ontological metadata structures.

The lack of human cognition within these ontologies can not eliminate unnecessary or ambiguous terms using results from previous analysis. This study concluded that users can query large datasets via the Internet more efficiently by eliminating these features within a self-learning model.

The role of metadata assessment can be seen in a variety of different fields. An Electronic Metadata Record (EMR), for example, is an emerging technology that is produced and edited when an electronic document is edited or created, such as a patient record or digital x-ray. A number of other related technologies for the medical industry have been developed to serve as a quality assurance and administrative tools. The process of accessing, viewing, and commenting on patient files or x-rays by physicians in electronic form can be documented and stored in a metadata file. Hardcopy records are often time consuming to complete, and they can be easily lost or destroyed. Thus, the ease of storing, accessing, and retrieving electronic metadata and files for medical data can help prevent litigation against malpractice lawsuits (McLean et al. 2008). Theodosiou et al. (2008), for example, developed a complex statistical analysis to retrieve biomedical articles from more than 4,800 journals to help support the decision making process. It is impossible to go through each of the 14 million individual manuscripts. Clustering and classification methods performed on metadata derived from traditional statistical techniques are used to explore and retrieve related information within biomedical literature. If properly maintained, metadata serves as a capable surrogate when querying scanned imagery or hard copy information is not feasible and futher validates in-situ decisions as they are reinforced by easily accessible support literature.

Metadata has the flexibility to capture many forms of qualitative and quantitative data such as numbers, text strings, domain values and dates. However, it does have its drawbacks. In addition to the time, resources and expertise required to populate the information, ancillary concerns exist. Metadata can be applied to any electronic resource, but there are data privacy concerns, especially within the medical community. McLean et al. (2008) discusses how metadata can be updated and collected to determine the number of times a medical professional has viewed patients' information within the EMR. Not only does this address privacy concerns by documenting access to particular records, but serves to report when, by whom and how long a digital record was viewed . In addition, EMR should not serve as an end all diagnostic tool, especially when clinical data do exist. Metadata should aid in the evaluation and decision making process. Skågeby (2008) used image sharing community to show this point. Metadata for the image (date of image, place, context, etc.) is collected and stored with the image. At what level is this technical and social metadata to be limited and from whom? Can certain tags be limited to a particular user's role such a private or public? Limiting these tags greatly reduces the amount of analysis that performed on the accompanying image, decreasing the availability to knowledge in order to make sound business decisions. As this applies to GIS metadata, a happy medium needs to be found so privacy concerns can be satisfied while dutiful analysis can be performed. Given the relative infancy of these subjects and lack of established doctrine, further analysis into these ideas is subject for further debate.

GIS and Metadata

The very nature of spatial data dictates that a different approach must be taken for assessment and reporting within the digital environment. The proliferation of spatial technologies underscores the widely accepted and legitimate role of metadata within the GIS user community (Limbach et al. 2004). All elements intrinsic to spatial data, such as those associated with position (e.g. latitude, longitude) as well as its representation (e.g., accuracy) must be carefully documented and recorded in GIS metadata. The creation of GIS data within the digital realm can take on a number of different forms. GIS data can be derived from larger data sets, created by digitization, produced using a Global Positioning Unit (GPS) or merely downloaded from the Internet (Chang 2008). It is important that information about the data format, a description of the data, the processes by which the data were created, the areal extent of the data and the people who aided in data creation be retained. Formal controls may dictate specific tolerances for horizontal and temporal accuracy. This information is not only important from a legal standpoint, but it also validates GIS analysis by speaking directly to such necessary components as its horizontal and temporal accuracy. GIS analysis is only as good as the data on which it is based. Metadata reinforces the data and ultimately the analysis and organizations which develop the GIS data.

GIS metadata is the formal structure in which this spatial information is stored for use by the larger GIS community. Since traditional GIS data are relatively recent, metadata standards must be flexible enough to adapt to new techniques. Policy should

dictate that these standards be revisited periodically to ensure adaptability that can be implemented through large-scale changes or the publishing of new metadata standards.

The GIS community has employed a set of content standards to ensure compatibility across the GIS community. The origins of a nationwide standard for GIS data began with the creation of the National Spatial Data Infrastructure (NSDI), signed by Executive Order 12906 on April 11, 1994. It states

> Standardized Documentation of Data. Beginning nine months from the date of this order, each agency shall document all new geospatial data it collects or produces, either directly or indirectly, using the standard under development by the FGDC (Federal Geographic Data Committee), and make that standardized documentation electronically accessible to the Clearinghouse network. Within one year of the date of this order, agencies shall adopt a schedule, developed in consultation with the FGDC, for documenting, to the extent practicable, geospatial data previously collected or produced, either directly or indirectly, and making that data documentation electronically accessible to the Clearinghouse network (FGDC 1998).

A Spatial Data Infrastructure (SDI) defines the elements for the implementation of a rudimentary geospatial information system. This includes the formal policies, fundamental data sets, hardware, software, and human resources required to collect, manage, access and render spatially related data at various political and administrative scales (Coleman and McLaughlin 1998). Nedovic-Budic et al. (2004) argued that GIS data and information transcends international boundaries through evidence that demonstrated similar data efforts in very different locations of the world (Australia and Illinois). It is difficult to dictate international data development efforts, but the need and

desire to share spatial data both nationally and internationally is placing importance on the use of multinational spatial data standards.

The Federal Geographic Data Committee (FGDC) is designed for creating and sharing geospatial data in the United States, but other nations have adapted this standard in part (FGDC 1998). The ISO (International Standards Organization) does maintain a metadata content standard for the use of GIS data worldwide, but portions of it are derived from its FGDC counterpart (Kresse and Fadaie 2004). FGDC metadata are saved in a format compatible with its editor. Acceptable software formats include HTML (Hyper Text Markup Language), XML (Extensible Markup Language), TXT (Text File) and SGML (Standard Generalized Markup Language). Because of its popularity, flexibility and application in the open source environment, this research will focus on the XML format. Batcheller (2008), Haas et al. (2003), and Tuchyna (2006) have all touted the use of XML as an advantageous format for the management of geospatial metadata because of its integration with the geography community. Geographers have devised a language called GML (Geography Markup Language) to express geographic phenomena. It uses syntax and grammar similar to XML. Consequently, converting between GML and XML is relatively simple (Kresse and Fadaie 2004). XML is a tagged format whose elements are composed of attributes and corresponding content. All popular GIS software packages have utilities that can import and export between XML and their native format.

FGDC metadata is divided into 7 sections or divisions that transcend descriptive, administrative and structural components. They include:

1. Identification Information
2. Data Quality Information
3. Spatial Data Organization Information
4. Spatial Reference Information
5. Entity and Attribute Information
6. Distribution Information
7. Metadata Reference Information (FGDC 2000)

Within these high-level divisions, subdivisions and eventually individual metadata elements can be populated to catalog various forms of information about the GIS data layer. The hierarchy of these divisions and subdivisions are standardized according to the Content Standard for Digital Spatial Metadata (CSDGM) produced by the FGDC. In addition to providing this structure, the FGDC also creates guidelines by dictating which metadata elements are to be populated. The FGDC requires that 7 metadata elements be populated for all GIS data. The FGDC also suggests that 15 metadata elements be populated. These elements are included in Table 1.

FGDC Required Elements	**FGDC SuggestedElements**	
Title	Dataset Responsible Party	Lineage Statement
Reference Date	Geography Locations by	Online Resource
Language	Coordinates (X and Y)	Metadata File Identifier
Topic Category	Data Character Set	Metadata Standard Name
Abstract	Spatial Resolution	Metadata Standard Version
Point of Contact	Distribution Format	Metadata Language
Metadata Date	Spatial Representation Type	Metadata Character Set
	Reference System	

Table 1. FGDC Required and Suggested Elements (FGDC 2000)

The above requirements represent a minimum baseline standard for metadata population. Organizations that share their GIS data or publish their metadata to large clearinghouses would most likely develop metadata standards above and beyond these minimum standards. Some of these non-suggested, but potentially important elements are captured in Table 2 (page 34) are and included in analysis as part of this research.

Studies revolving around FGDC content standards show that they are robust enough to support most forms of GIS data. Nonetheless, the FGDC actively creates content standards for new technologies and manners in which GIS data are collected. One such example is the FGDC content standard for Remotely Sensed Data. Instead of 7 basic divisions, Remotely Sensed metadata are captured in 9 major divisions. This includes 2 divisions germane to the equipment and methods used to capture the raster data, in addition to the 7 existing divisions described above. This additional information specific to the remotely sensed imagery includes the platform name, sensor information and algorithm information (FGDC 2002). Please refer to Chapter VI for further elaboration.

Although metadata's original use in the beginning was simply as a means to catalog data, it has become a science in itself. Even early research and commentary on the concept of metadata has touted its value as an effective decision making tool, regardless of its native format (Wong and Yu 1996). David Lanter was one of the first pioneers in the field of metadata and its study as a separate field as opposed to an extension of GIS. Lanter (1993) explored methodology to integrate spatial data to a stand alone database long before metadata was stored in this format. This was done before

mainstream GIS software was able to edit and handle the different forms of metadata. Before the introduction of ESRI's ArcCatalog 8.0, GIS metadata was edited and managed in standard text editors such as NotePad or some propriety application such as CorpsMet created by the Army Corps of Engineers, NJMetaLite by the New Jersey Department of Environmental Protection or the NOAA FGDC Metadata Toolkit. Lanter made contributions in groundbreaking software by creating an application that could glean spatial information that could be populated into metadata.

Even before more advanced tools could populate GIS metadata elements as they do now, Lanter (1994) looked into compiling statistics about metadata elements within the confines of ESRI command line software. Regardless, one recurring theme with metadata is the time and energy required to maintain it. Leiden, et al. (2001) showed that the population of geospatial metadata is a monotonous process and therefore subject to error. Because of this, Doctorow (2001) maintains that human nature alone undermines the immediate and long-term goals of metadata for an organization and the GIS user community. FGDC allows more than 400 individual metadata elements to be entered by the user. While the omission of one minor element would not degrade a layer's metadata or invalidate the GIS data on which it is based, the FGDC has determined, required and recommended metadata elements.

Although agency based controls and requirements are dictated by the FGDC, enforcing them at the program level must take a bottom-up approach. Batcheller (2008) explored ways to streamline and automate the creation of GIS metadata, but does so within the confines of Dublin Core using a Dynamically Linked Library (DLL) for

Environmental Systems Research Institute (ESRI) software. Batcheller's work is probably closest to the line of research conducted in this dissertation, but his work is limited to both a platform environment and software package. Users must use the appropriate software package, in this case ArcCatalog. In addition, users are limited to the Windows Operating System. Lastly, issues regarding patches, software updates and versioning must be reconciled so these DLLs can run properly under these conditions. That leaves a lot of the GIS user community unable to repeat or take advantage of these results. Extending the scope and scale of Batcheller's work to all computing environments serves as the practical impetus for this work. Parallel to this ideal, ways to encapsulate, analyze and express this information using various technologies serves as the intellectual motivation for this research. This crosswalk discussed by Batcheller between the Dublin Core metadata standard falls outside of the scope of this work, but explores interesting aspects of GIS metadata management methodology and feasibility of extending programming concepts to the field of metadata. Consequently, this research intends to assess and report existing metadata outside of the confines of a specific software package, thus increasing usership with less reliance on particular software platforms.

GIS and Statistical Programming

The use of statistical programming techniques within the realm of GIS is an evolving technology. While fields such as geomatics and geodesy have developed from the propagation of GIS, related subjects can be developed from statistics and the digital use of statistical programming. One of these fields is data mining. The principles of data

mining are borne from the fact that increasingly large databases and data sets cannot be assessed and visualized using traditional techniques. As a result, data mining is sometimes referred to as *Knowledge Discovery in Data* (KDD), or the "discovery of interesting, implicit and previously unknown knowledge from large databases" (Koperski, et al. 1996). Advances in computing technologies have allowed only important and relevant information to be gleaned and processed from these data. The one problem with applying traditional data mining techniques such as linear regression to GIS data is determining how these data are spatially autocorrelated (Kazar 2004). Thus, there are advantages and disadvantages of merging GIS with the concepts of data mining.

Precursory to the implementation of data mining is the notion of Exploratory Data Analysis (EDA). EDA techniques are used to visualize relationships and dimensions of data, in this case information gleaned from GIS metadata. The applications of EDA to GIS data and metadata are especially useful because it allows various dimensions of data to be viewed and analyzed. This is done for a variety of reasons, which may include pattern detection and hypothesis formulation before performing any data mining or generalization of that data. Ultimately, EDA allows users to get accustomed to the data and perhaps guide the data mining process using data in its purest form (Hoaglin et al. 1983).

EDA uses graphical techniques such as scatter plots, run charts, Pareto charts and stem plots to determine how the data interacts with itself, other variables or time. The integration of EDA with GIS has been a relatively new phenomenon. One of the first applications of EDA techniques to spatial data was done by Bao et al. (1995) in the mid

1990s. Originally, EDA also included spatial statistical techniques such Moran's I and Geary G. However, now spatial statistics is now regarded as a separate subject and treated accordingly by most of the GIS community. One of the biggest challenges to these innovators was the marriage of these statistical techniques to spatial information, measuring how space matters and rendering this information in a relatively rudimentary digital environment.

Later, Dramowicz and Dramowicz (2004) used EDA to view statistical distributions of data to be displayed using a choropleth mapping technique. The goal of any thematic map is to clearly display as much information as possible. However, these raw and derived data can be grouped a variety of different ways, such as the equal interval, quantile, standard deviation and natural break classification scheme. EDA can be used to determine what classification scheme will work best for the data. The data are not going to change. EDA can be used to help determine an adequate approach to working with the data.

The first uses of full-fledged data mining within spatial data did not occur until the end of the 20th century with Aref and Samet (1991) and Bell, et al. (1994). In fact, it was not even related to terrestrial data, as Bell used data mining procedures within raster imagery patterns to identify volcanoes on the planet Venus. Closer to home, prevalent literature has shown that spatial data mining has been most closely related to addressing transportation-related problems by Yao (2007). Fundamental divisions of transportation infrastructure can be divided into nodes, vertices, edges and networks that can easily be represented within the digital environment through the concept of topology. Some GIS

data formats can enforce topological relationships among and between different features. Yao's 2007 study used spatial data mining techniques to explore deficiencies in public transit, using Atlanta, Georgia as a case study. Grubisec and Zook, in 2008, scaled out their application to explore the financial and physical accessibility of air travel to the American populace using spatial data mining techniques.

That is not to say that environmental and sociological applications see no use in data mining as a problem-solving tool. However, other considerations need to be taken into account, most notably scale and the use of the enumeration unit. Spielman and Thill (2008), using 79 different attributes, explored a number of methods to map New York City. Many phenomena transcend New York's 2217 census tracts, bringing to light the use of a self-organizing map (SOM) created through data mining techniques. However, this study found that the SOM has as many limitations as the arbitrarily drawn census tract boundary in this case. Another example of data mining in a geographic context is Su et al. (2004), who used data mining techniques to correlate fish behavior with environmental conditions. However, the fundamental analysis unit to express fish behavior is the cell grid. Representing this behavior in the digital environment is a difficult challenge in itself. Lastly, another challenge within the social realm exists when dealing with the concepts of mining spatial information. While the focus of this research will be on the textual elements used to describe the data set and methodology to assess and report this information, this information can easily be applied in a spatial context. Taipale (2007), for example, spoke of the dangers of trying to find a 'reconciliation' between using data mining as an effective educational or even counter-terrorism tool and

the personal freedoms of those whose information is being mined. Regardless, as data mining techniques are more readily infused into the spatial sciences, we as a community must be cognizant to ensure that little transparency of individual information exists. As this applies to GIS metadata, identifying information should be kept at the program level or database level.

The Technology Acceptance Model (TAM)

While the intersection of these various subjects serve as the theoretical impetus of this research, assessing these techniques can take on a variety of different forms. How will technology be further disseminated in the working world? How can this technology be assessed? While factors such as generated income and usership are quantitative in nature, they are interwoven with determinants such as marketing, depth and level of human-computer interaction, organizational structure and management of the technology, which are tangential at best to this technology. Some of these factors do not speak to the effectiveness of the technology, but the diffusion of this technology which helps to proliferate its use. With this 'chicken or the egg' scenario, it is sometimes difficult to compartmentalize a valid measurement scale to assess technology acceptance alone for a single piece of technology within the user community.

Given that the intended usership of this research is geared towards GIS professionals as opposed to developers or programmers, a testing mechanism geared toward this group would be more appropriate than testing code efficiency or complexity. While some of these latter topics will be mentioned in the Results section of this dissertation, the Technology Acceptance Model (TAM) has served as a means to assess

and quantify the effectiveness of a technology for almost 20 years and will do so once again for this research. The TAM that we know of today, developed by Davis (1989) was originally created as a means to universally quantify the effectiveness of technology. It was born from the fact that the adoption of new technologies is dependent upon such ambiguous notions as psychological disposition, attitudes, intentions and our own personal biases related to this new technology that make it difficult to test and validate (Bagozzi et al., 1992). TAM is actually the technical manifestation of the TRA (Theory of Reasoned Action) first proposed by Fishbein and Ajzen (1975). TRA is the theory in which beliefs, composed of attitudes, values and opinions at the individual level, eventually result in enacted behavior. Within the TAM, this enacted behavior is the decision to adopt technology.

Davis used empirical studies to find that a technology's acceptance is most related to its 1) Perceived Usefulness and 2) Perceived Ease of Use. Perceived Usefulness refers to the quality in which a technology would help one's job performance. While others such as Stewart (1986), Shultz and Slevin (1975) and Robey (1979) did explore this usefulness dimension, TAM also looks at this in concert with this technology's "freedom from difficulty or great effort" (Davis 1989). This ease of use factor helps support the self-efficacy theory supported by Bandura (1982). While Bandura does distinguish between the roots of this effectiveness and seminal outcomes which at times can be paradoxical, Davis encapsulates this within one encompassing desired end-state of ultimately accomplishing one's tasks with as little effort as possible. Studies have actually shown the relationship between this Perceived Usefulness and Ease of Use with

the adoption of technology is irregardless of variables such as gender and computer experience (Dimitrova and Chen 2006).

Using these two indicators as a guideline, Davis creates questions that try to explain the usage and acceptance patterns of a technology as per TRA. Users of the technology are asked to scale responses to these questions similar on a 7 point Likert-type scale, representing "Strongly Agree" through "Strongly Disagree". Regression analysis between this effectiveness and ease of use variables is determined at various confidence intervals. In addition, principal components analysis is used to explain the variance of usage intentions as a function of Perceived Usefulness and this attitude towards the technology. This and other hypotheses related to Perceived Usefulness, Perceived Ease of Use, Attitude Towards Using the technology and behavioral intention of use are tested among each other (Masron 2007).

TAM represents a milestone towards understanding human behavior as applied to the technology realm. Germane to this research, Masron (2007) applied TAM to e-learning while Koufaris (2002) tested TAM within the e-commerce environment. Koufaris integrates technology and human behavior when applied to online shopping. Visiting an online store for the first time has serious consequences on whether the visitor will visit again or make unplanned purchases. Making this online shopping experience an easy, enjoyable and memorable one is of utmost concern to these businesses. The order in which material is presented, the amount of material presented and the user's cognitive impression of this material, its volume and its underlying messages play into

this perceived enjoyment factor. Finally, all of these facets need to be assessed in a manner free of bias and misconception.

However, TAM does have its limitations. Hufnagle and Conca (1994), among others including Davis himself, feel that TAM does not adequately explain for social influences. More specifically, it is difficult to discern whether intent, attitude or some other referent characteristic sufficiently explain usage behavior. Principal Component Analysis can only do so much within the paradigm of the testing environment. In addition, it is difficult to explain how this physiological attachment related to attitude and behavioral intention can be assessed within TAM. Malhotra and Galleta (1999) do attempt to explain this by expanding the dimensionality of testing elements within a rotated component matrix. However, any technology is an investment. Even using this model, it is difficult to realize the value of the large-scale investment for an organization at an individual level given the multiple intrinsic behaviors and intentions independent of this organizational goal.

CHAPTER III

METHODOLOGY

GIS metadata serves as the formal means by which spatially-related phenomena can be catalogued within a formal framework. It is here where tacit information can be codified for use by the larger GIS community. Given the ever-increasing size of GIS data sets and the proficiency with which GIS data are created, there needs to be a mechanism to assess the quality of these data not seen in previous generations or documented in literature. Programming techniques and software packages have allowed users to assess information that would take a human days or perhaps weeks to do. Applying these techniques to the field of GIS metadata is the cornerstone of this research. While devising these techniques, this research addressed the following issues:

1. Using computer assisted data mining techniques, what dependencies or trends were found not readily apparent within GIS metadata?
2. How has EDA (Exploratory Data Analysis) helped users view information about their GIS metadata?
3. What statistical metrics were gleaned from GIS metadata to best judge the validity of a GIS data set for large-scale data analysis?
4. What deficiencies existed within current GIS metadata constructs undermine its ability to be assessed?

In doing this analysis, dependency on particular software platforms or packages must be minimized. Replication by the larger GIS user community can be done via these techniques. This lack of dependency was one of the larger benefits that can be taken from this body of work. Applying open-source programming techniques to something as specific as GIS metadata can help perpetuate the democratization of free and inexpensive and statistical software. Software for the geographic and statistical sciences can be expensive and platform dependent. Applications covered in this research may require more programming knowledge, but can do so in more diversified software environments. Organizations such as schools or public agencies that lack the resources to purchase expensive site licenses can employ open source tools to accomplish many of the same tasks as their counterparts in the private sector.

It is difficult to quantitatively judge GIS data quality because many fields within GIS metadata are nominal in nature. While FGDC and agency policy can tell if a particular field is populated, it is impossible to tell if it is correct. However, temporal accuracy (difference between the current date and the publication date) and horizontal accuracy are quantitative measures that speak directly to a user's willingness to implement the data set. In addressing the first question above, it was difficult to discern trends from GIS data from a variety of different sources, contacts, contractors and parts of the world. However, data mining techniques used on this information extracted interesting relationships between the dimensionality and cardinality of the data that would otherwise go unnoticed by the human eye.

However, any form of data mining, whether it is machine learning, clustering or rule induction/association mining, must be validated by the human component. For example, deriving an association from GIS metadata may dictate that data with a better temporal accuracy (newer) also have a better positional accuracy. This may be intuitive because technological constructs may allow for better data to be created. However, machines do not know that. All associations or predictions must be ultimately validated by the human component to ensure its worth. These relationships can be proven within statistical significance levels using data mining techniques (Klösgen and Żytkow 2002). Given the use of varying and integrated software techniques, the methods used in this work were subdivided logically based on these techniques. This holistic process is highlighted in Figure 1 and described in the following sections.

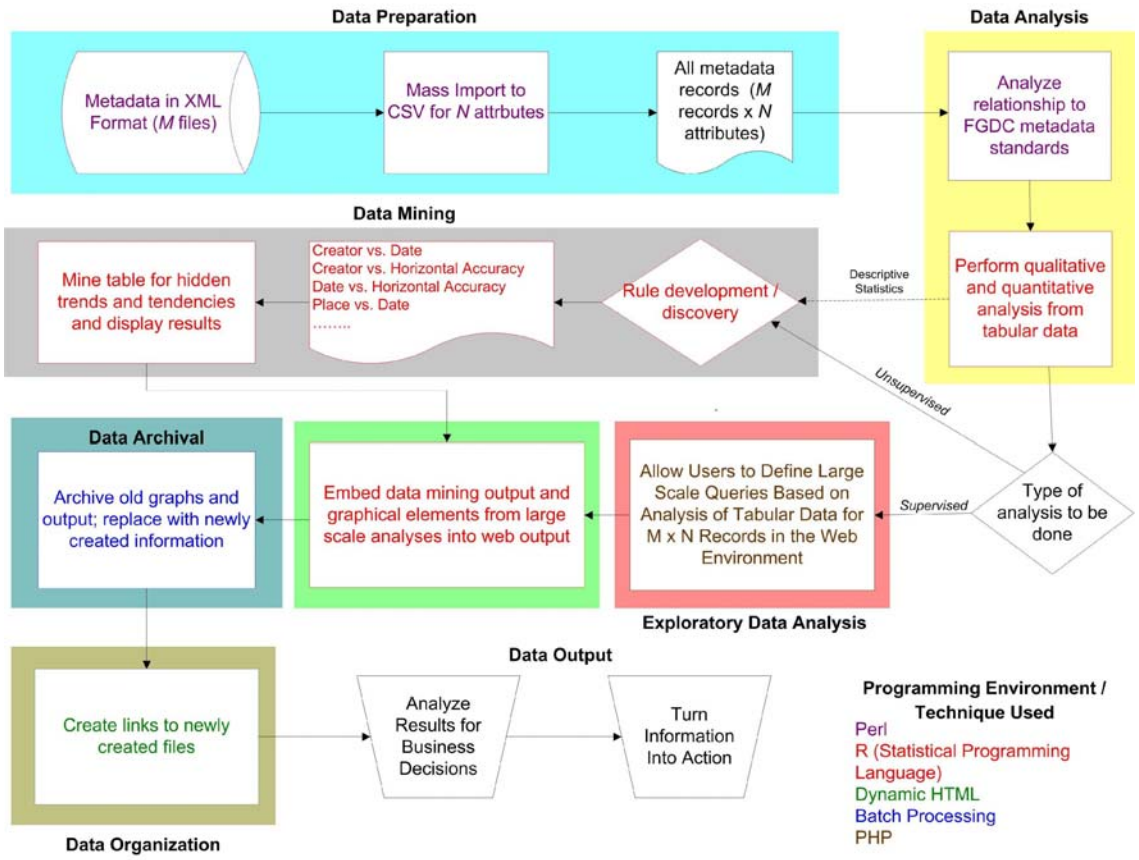

Figure 1. Proposed Flow Diagram for MART

MART (Metadata Assessment and Reporting Tool)

The assessment of GIS metadata, application of data mining techniques and large-scale user-defined queries of compartmentalized metadata information serve as the theoretical impetus of this research. However, if these theories can not be applied and practiced in the real world then they have little utility to Geographic Information Science. Applying these theories to current Information Technology (IT) infrastructure served as the technical or practical impetus of this research. By doing this, IT and GIS

professionals can identify limitations of current technological constructs as it applies to GIS metadata assessment and work on ways to better quantify and render this information in the computing environment.

Given the popularity of the Internet as a means to deliver high volumes of information in an organized manner via the digital environment, this practical impetus used the Internet as this assessment and ultimately rendering medium. This tool is called MART (Metadata Assessment and Reporting Tool) and is composed of different modules. These modules used open source programming tools to interface with the GIS metadata, assess the metadata, graph the metadata, create output in a format compatible with the Internet and work in concert with a web server to deliver this information over the Internet. These techniques represented programming environments and the media used to store and process the GIS metadata in its various forms. They are highlighted in Figure 1. The modules within each color-coded section represent a particular programming language discussed in the following chapters.

The use of open source tools was the preferred method for application delivery in MART because open source tools can run on most operating systems and digital environments. It was impossible to gauge server configurations for potential users. Therefore, using proprietary tools catered to specific GIS applications (e.g. ArcObjetcts), operating systems (e.g. Visual Basic) or web browsers (e.g. Active Server Pages) limited potential users because of these software limitations. Furthermore, rectifying issues of versions and Service Packs that plague the typical Windows environment further restricted users. These issues will be addressed in Chapter VI.

Given the intended users of this application, prior knowledge of open source programming languages is not required. These different modules were tied together using a batch file and were scheduled to run with normal operating systems functions such as backups, virus scans and report generation. MART can be scheduled on a weekly basis when activity is low. The reports generated were linked to raw data files, graphs, text and tables embedded within HTML files generated upon deployment of this application. Little user interaction with MART by the users was required.

Database Selection

The GIS metadata used in this project come from a variety of different sources. The author previously worked for a GIS Support Center that served as a central repository for government GIS data throughout the world. Assessing the quality of these data, including the metadata, serves as the impetus for this proposed course of study.

The author has 112 GIS databases to work with, each of which contains approximately 43 – 116 individual data layers. The 112 databases represent different locations throughout the world. They were created by different people, contractors and at different times. As a result, they represent a cross-section of the experience, competence and aptitude of the GIS user community and the population of the appropriate metadata. In addition, the researcher has access to hundreds of other data layers that can serve as an adequate sample to test in MART. While there are no security concerns associated with these data, at no time will the GIS data be accessible within the confines of this project. Each of these layers contained a separate metadata file, and combined with the other

metadata metrics from the same database, one discerned specific trends about the information in the database.

Forty users were selected to serve as a test bed for this application. Most of these GIS data resided in personal geo-database (PGDB) format popular with ESRI software packages. However all metadata will be converted to XML format for further processing in the open source environment. Please refer to the Discussion for remediations and issues associated with this process.

Data Preparation

Mining and analyzing information required it to be in some organized format based off of some existing standard. Extensible Markup Language (XML) formalizes the FGDC compliancy within a nested format that allows for easy and simple navigation within the 7 major sections of GIS metadata.

XML in its Native Format

Information about individual data layers resided in XML format. These XML files existed separate and independent of each other. As a result, each XML file was opened, accessed and the information written to the single data file that was analyzed. This XML file was then closed so the next XML file can be opened and accessed.

XML is a format where the tagged markers (< >) qualify and organize the information outside of the tags. An example of XML code can be seen in Figure 2. One can see that the value for the tag named *pubdate* is 20050401, the FGDC metadata

convention for the date April 1st, 2005. This *pubdate* tag is a smaller part of the *citeinfo* object because the terminator (</citeinfo>) lies after the *pubdate* terminator.

```
<citation>
    <citeinfo>
        <origin>Tele Atlas North America, Inc./Geographic Data Technol
        <pubdate>20050401</pubdate>
        <title>U.S. Census Block Groups</title>
        <ftname Sync="TRUE">blkgrp.sdc</ftname>
        <geoform Sync="TRUE">vector digital data</geoform>
        <edition>2005</edition>
        <serinfo>
            <sername>ESRIÂ® Data &amp;amp; Maps</sername>
            <issue>2005</issue>
        </serinfo>
        <pubinfo>
            <pubplace>Redlands, California, USA</pubplace>
            <publish>ESRI</publish>
        </pubinfo>
        <othercit>Location: \usa\census</othercit>
        <origin>ESRI</origin>
    </citeinfo>
</citation>
```

Figure 2. Example of XML Code

The location of these tags and terminators dictates a hierarchy of objects within GIS metadata and serves as a means to access these objects programmatically in a systematic manner.

The most current GIS metadata standard (ISO 19115/FGDC CGDSM Version 3.0) allows for more than 400 individual elements to be populated (Kresse and Fadaie 2004). Within an XML file, there exist different types of qualitative and quantitative information such as strings, text fields, integer numbers and floating point numbers that transcend the various measurement scales (nominal, ordinal, interval, ratio) accessible through these tags. Using the FGDC metadata documentation as a guideline, 43 metadata attributes were collected for each data layer for further analysis. These attributes

included the 7 required and 15 recommended fields as per FGDC guidelines (FGDC 2000), in addition to another 21 fields that the author felt may contain useful information. While dependencies do exist between these elements (a point of contact should have a phone number, address, phone number, etc.), any additional fields are not collected. As a result, only 7 elements were required in these analyses. Using this as a starting point, these 22 components and corresponding parts should serve as a minimum standard for GIS users. Business practice or organizations may dictate standards above and beyond what FGDC dictates. As a result, other fields were selected for use in further analysis. Further elaboration of these fields is discussed in the Methodology. These attributes are highlighted in Table 2 below.

Table 2. Name of all Components Collected for Each Record in MART

This information is stored in a hash element, a Perl construct that references related memory locations using the hash name described below. These values are stored a CSV file for use in the data analysis and mining process. The FGDC-required and suggested elements (prefixes r and s) are checked in Perl to ensure they are populated and statistics are computed to summarize FGDC compliance. Non-required elements that may of interest to the research are also collected from metadata and denoted by the n prefix

Hash Name	Definition	Explanation
Current_Date	Date that the information is collected	Collected when the program is run
attributes	Name of array used to store attributes	Memory location
fileName	Name of XML file being assessed	To be used for display purposes
lat1_s	North Bounding Latitude of extent rectangle	Extracted from metadata – used to determine location
lat2_s	South Bounding Latitude of extent rectangle	Extracted from metadata – used to determine location
Geographic Location	East Bounding Latitude of extent rectangle	Used as a check to ensure these are populated. If one value is populated, they all are
long2_s	West Bounding Latitude of extent rectangle	
n01_areaofextentsqmi	Area of map extend in square miles	Non-Required Element
n02_areaofextentsqkm	Area of map extend in square kilometers	Non-Required Element
n03_updatefrequency	How often the data should be updated	
n04_placekey	Name of place	Extracted from keywords
n05_geoid	Name of Geoid used for projection	Should be populated if projection information has been adequately populated

n06_ellipsoid	Name of Ellipsoid used for projection	Should be populated if projection information has been adequately populated
n07_semimajor	Semi-major axis distance used for projection	Should be populated if projection information has been adequately populated
n08_flattening	Flattening ratio used in projection information	Should be populated if projection information has been adequately populated
n09_sdorganization	Mechanism used to represent spatial information in the data set	Typically vector or raster
n10_sdtstype	Encoding format according to SDTS standards	Approximately 20 different values
n11_objectcount	Number of individual objects composing GIS data set	Integer value
n12_attdefsystem	How the attributes have been defined	Could be ESRI or another organization
n13_contactorganization	Contact organization associated with required contact person	Non-Required Element
n14_mdorganization	Metadata organization associated with required metadata	Non-Required Element
n15_contactposition	Contact position associated with required contact	Non-Required Element
n16_mdposition	Metadata position associated with required metadata contact	Non-Required Element
n17_xdistance_mi	Average east-west distance of extent rectangle in miles	Computed using Perl GIS modules
n18_ydistance_mi	Average north-south distance of extent rectangle in miles	Computed using Perl GIS modules
n19_xdistance_km	Average east-west distance of extent rectangle in kilometers	Computed using Perl GIS modules
n20_ydistance_km	Average north-south distance of extent rectangle in miles	Computed using Perl GIS modules
n21_useconstraints	Any constraints that exist with regards to data use	Non-Required Element
n22_accessconstraints	Any constraints that exist with regards to data access	Non-Required Element
n23_numattributes	Number of attributes used to describe the attribute table	Non-Required Element
r01_Data_Set_Title	Title of data set	FGDC Required Element
r02_Publication_Date	Publication data for date	FGDC Required Element
r03_Language	Language of data set	FGDC Required Element
r04_Data_Theme	Data set theme keyword	FGDC Required Element
r05_Abstract	Short description of data set	FGDC Required Element
r06_Metadata_POC	Person who completed the metadata	FGDC Required Element
r07_Metadata_Date	Date that metadata was completed	FGDC Required Element
rCount	Number of possible required elements for a data set	Usually 7, but can be appended when requirements change
rCountFound	Number of FGDC-required elements found for a data set	Used for computations in report.html

rMissing	Names of FGDC-required elements that were not found	Used for output in report.html
s00_Spatial_Resolution	Horizontal accuracy extracted from horizontal accuracy statement	FGDC-Suggested Element
s01_Distribution_Format	Information about the distributor of and options for obtaining the data set	FGDC-Suggested Element
s02_Additional_Spatial	Vertical accuracy, if applicable	FGDC-Suggested Element
s03_Spatial_Representation	How spatial data are represented	FGDC-Suggested Element
s04_Reference_System	Highest level of projection information	FGDC-Suggested Element - could be UTM, State Plane, etc.
s05_Lineage_Statement	Processing that has been performed on the data set	FGDC-Suggested Element
s06_Online_Resource	Physical location of database	FGDC-Suggested Element
s07_Metadata_Field	Online location of standard described	FGDC-Suggested Element
s08_Metadata_Standard_Name	Name of metadata standard	FGDC-Suggested Element
s09_Metadata_Standard_Version	Version of metadata standard	FGDC-Suggested Element
s10_Metadata_Language	Language of metadata	FGDC-Suggested Element
s11_Metadata_Character_Set	Character set used by metadata	FGDC-Suggested Element
s12_locationlong	Longitude of area centroid	FGDC-Suggested Element - 1 component of location
s13_locationlat	Latitude of area centroid	FGDC-Suggested Element - 1 component of location
s14_Responsible_Party	Name of contact for the data set	FGDC-Suggested Element
s15_Data_Set_Character_Set	Character set used by data set	FGDC-Suggested Element - default is UTF-8
sCount	Number of possible suggested elements for a data set	Usually 15, but can be appended when requirements change
sCountFound	Number of FGDC-suggested elements found for a data set	Used for computations in report.html
sMissing	Names of FGDC-suggested elements that were not found	Used for output in report.html

In order for knowledge to be properly applied, a simple tabular structure consisting of horizontal elements (records) and vertical elements (attributes) is more than acceptable (Klösgen and Żytkow 2002). The method of accessing and extracting multiple XML files for necessary information and placing it into an acceptable tabular format was done via the Perl programming language. Perl is short for *P*ractical

*E*xtraction and *R*eporting *L*anguage and is typically used to perform string manipulations and extract information for strings or large textual data sets. Perl is open-source, meaning that it can run on a variety of operating systems and is freely accessible to the computing community. It is able to run without any special downloads or extensions (Schwartz et al. 2005). The use of Perl in this application was to extract required FGDC metadata elements from XML metadata and place them into a large table so these data can be processed. Perl also checked to make sure certain metadata elements have been populated, noting the results and writing them to a web page. Perl has capabilities to traverse XML schema as shown in Figure 2 and placed these metadata attributes into tabular format for part of the data analysis and data mining portion of this project.

Data Pre-Processing

Before doing this process, however, some rudimentary checks and balances were performed by the program. Perl runs at a command line, meaning that interaction or deploying the code occurs with text commands written at a command line. For the Windows computing environment, this occurs at a DOS (Disk Operating System) prompt or a shell script in Linux. An example of this DOS command is as follows:

```
C:\PERL_Test>perl extract_csv.pl -s c:\gis_metadata\ -o
c:\gis_output\test.csv -r c:\gis_reports\fgdc_report.html
-t c:\datamining\transaction.txt
```

The Perl program was deployed using the `Perl extract_csv.pl` command, with `extract_csv.pl` being the name of the Perl program. Import parameters were also created that can be changed by the user. These parameters were:

-s (source data) – Folder location where GIS metadata in XML format resides

-o (output file) – Location and name of CSV file that will be populated in the data preparation process

-r (report file) – Location and name of the HTML file that will be dynamically created to report FGDC compliancy

-t (transaction file) – File to be used for association rule learning

If multiple or periodic (weekly, for example) executions of this data preparation component were needed, these commands can be placed in a batch file and executed at a scheduled time. Most operating systems have a scheduler utility that can run commands at a specified time. If no parameters were entered, default values are used. The default value for the source data was the current folder where the Perl program is located, the name of the default CSV file is 'test.csv' and the name of the report file is 'report.html'. The CSV and HTML files would be written to the current folder where the Perl code resides if no options are specified.

In addition, the source information which showed where the XML files are located can be generalized or highly specific. If the user wishes to assess all GIS data located on a particular drive, the user can highlight that. If a user wants to assess only GIS data for a specific project, the –s parameter can be specific such as:

```
C:/GIS_Data/North_Carolina/Greensboro/Jan09BikePaths
```

After determining these input parameters, this code performed a couple other checks to prevent any premature termination or logical errors within the code. First of all, the code checked to ensure that the CSV and HTML files can be opened. If the CSV files already existed, then it needed to be opened for appending. Otherwise, this file was opened for writing. In addition, if these files have a special lock or permission applied to them that would prevent this writing, the program terminated on the spot while indicating the problem with an error message.

Another problem would present itself if the CSV file already exists and is opened for appending. One possible scenario may be that additional XML files were added to the source folder and the data need to be re-prepared. Therefore, if this existing CSV file was opened for appending, duplication of records may occur if safeguards are not employed. As a result, a subroutine within the Perl code checked the file name and date to see if a current record for this file existed. If a record did exist for that XML file, it will merely overwrite the existing record instead of having 2 separate instances for the same XML file within the CSV file. If the record did not already exist, there will be no need to overwrite any existing records and the record was appended to the existing list of records extracted from the XML metadata file.

Data Processing

Perl employs a number of data structures so large and related data sets can be dynamically accessed and added. After these precursory checks and the appropriate files were opened for editing in this application, the programmer employed a *hash object* to place all extracted metadata information into one information structure that is referenced by an applicable variable name (Schwartz et al. 2005). A hash object is a simple, but higher level alternative to more complex and resource intensive data structures such as dynamic arrays, linked lists and binary trees. The hash object in this application was named *metadata*. It is a lot like an array, but instead of referencing memory locations using a numerical index (0, 1, 2, 3, etc.), it does so by names. Examples of these names are *r02_Publication_Date*, *s04_Reference_System* and *n22_accessconstraint.* The prefix *r* was denoted by the researcher to designate that this metadata value is required by the FGDC. The prefixes *s* and *n* represent FGDC-suggested and non-required features, respectively. These names and conventions are much more intuitive than integer numbers. The non-required features were extracted from the metadata to increase the cardinality and dimensionality of the data mining analysis. A list of these hash elements, representing a list of all elements extracted from each metadata file, and a short explanation of each is in Table 1. An example of this output is shown in Table 3 in while this Perl code is highlighted in APPENDIX C.

Table 3. A Sample of 12 XML Metadata Files

This is an example of the CSV output. Each record, an XML file containing information for 1 GIS data layer, is represented by a hash element in Perl. Seven of these hash elements are FGDC required metadata elements while 15 of these elements are suggested by the FGDC. Sixteen suggested elements were actually extracted, but 2 hash elements (latitude and longitude) were used to ascertain the location requirement. Another 21 elements were also extracted or derived for use for data analysis and mining purposes. The other hash elements contained within this file support the analysis and suggested elements. The hash element was emptied after the FGDC-compliancy calculations are performed and the CSV value has been populated.

File_Name	Current_Date	n01_areaofextentsqmi	n03_updatefrequency	n04_placekey	n05_geoid
control_point.xml	20080716	'7.71197865088604'	As needed'	'Maryland'	'D_WGS_1984'
elevations.xml	20080716	'804.776061637777'	'As needed'	'Indiana'	'D_WGS_1984'
extent.xml	20080716	'409.627173068173'	'As needed'	'Indiana'	'D_WGS_1984'
hospitals.xml	20080716	'0.011441709633544'	'Irregular'	'Kentucky'	'D_WGS_1984'
lakes.xml	20080716	'6.34903447257748'	'As needed'	'Nevada'	'D_WGS_1984'
landing_zone.xml	20080716	'160.506486829238'	'As needed'	'Kentucky'	'D_WGS_1984'
military_range.xml	20080716	'141.465097159135'	'As needed'	'Kentucky'	'D_WGS_1984'
streams.xml	20080716	'1.11540862831335'	'As needed'	'Nevada'	'D_WGS_1984'
target_line.xml	20080716	'67.508374718607'	'As needed'	'Kentucky'	'D_WGS_1984'
towers.xml	20080716	'97.7402009360382'	'Irregular'	'Kentucky'	'D_WGS_1984'
training_area.xml	20080716	'261.610695148978'	'As needed'	'Kentucky'	'D_WGS_1984'
usgs_quads.xml	20080716	'103215.28965907'	'As needed'	'Maryland'	'D_WGS_1984'

Using the data set title as an example, the hash declaration was originally set to the path of the intended XML object. Using the example in Figure 2, one needed to access the `<title>` tag first through `<idinfo>`, `<citation>` and the `<citeinfo>` tags, respectively. This path, separated by the underscore (_) is the original value of the hash element.

```
$metadata{"r01_Data_Set_Title"} => "idinfo_citation_citeinfo_title";
```

After decomposing this path and traversing it within the XML file, it set the value of this hash object to be whatever is between the appropriate tags, which in this case is "U.S. Census Block Groups".

```
$metadata{"r01_Data_Set_Title"} = “U.S. Census Block Groups”
```

This value was populated into the spreadsheet in CSV format. If no tag existed or the tag was empty, it means that this metadata element has not been adequately populated and a value of NOT FOUND is entered into the appropriate record-attribute pair. During the population of the CSV table used for analysis and data mining, counters were used to note the number of required and suggested FGDC elements found versus the number of occurrences of NOT FOUND. These results were dynamically written to a HTML file on a record by record basis, noting what records or layers have FGDC-compliant metadata. While statistical programming languages would be needed to make complex calculations, Perl performed this basic analysis in ensuring FGDC compliancy. It checked to make sure that the appropriate FGDC compliant elements are populated. However, it is impossible to tell if these populated values are correct – i.e. the metadata values reflected what they really should represent. A hash element was also created and incremented to count the number of FGDC compliant metadata features on a record by record basis. Populating the hash elements and checking their validity within one loop cut down on the complexity of the program and therefore the run-time of the program. The number of FGDC compliant layers and individual features were summarized within the confines of this web page as seen in Figure 3.

FGDC Compliancy Report

File Name	Layer Name	Required FGDC Features	Suggested FGDC Features	Missing Features
./control_point.xml	Monumented Benchmarks, BG Thomas Baker Training Site (Lil Aaron Strauss)		14	Metadata Standard Version
./elevations.xml	20 Meter Elevation Contour Line, Fort Knox	6	14	Metadata POC, Responsible Party
./extent.xml	Map Extent, Fort Knox			NONE
./hospitals.xml	NOT FOUND	6		Data Set Title
./lakes.xml	Water Body Areas, Stead Training Site			NONE
./landing_zone.xml	Military Landing Zone, Fort Knox		14	Lineage Statement
./military_range.xml	Military Range Area, Fort Knox			NONE
./streams.xml	Water Course Centerlines, Stead Training Site		14	Lineage Statement
./target_line.xml	Military Target Line, Fort Knox	6	14	Publication Date, Responsible Party
./towers.xml	Tower Area, Fort Knox			NONE
./training_area.xml	Training Area, Fort Knox			NONE
./usgs_quads.xml	1:24,000 USGS Quadrangles, Maryland and West Virginia			NONE

9 out of 12 layers (75.00%) had all of the FGDC Required metadata components
81 out of 84 individual FGDC required elements (96.43%) were adequately populated

7 out of 12 layers (58.33%) had all of the FGDC Suggested metadata components
175 out of 180 individual FGDC required elements (97.22%) were adequately populated

Figure 3. Sample Output for FGDC Compliancy Report

Since all of the output in this report can be dynamically created using HTML and requires no graphical output, this report was created using Perl. Graphical output was created using another programming language called R and synthesized with this output within the confines of a web page. While VBA/ArcObjects programming techniques can programmatically access the appropriate metadata elements and statistical software such as Excel, SPSS/SAS (for later processing), this code uses open source programming technologies. In this way, it can run under any operating system and computing environment. An infinite number of metadata files can be read. Twelve were read in Figure 3 so the results could be easily displayed.

After all attributes for a particular record were extracted from the XML file and populated into spreadsheet format, the next XML file was searched. The hash element

was emptied after the FGDC-compliancy calculations are performed and the CSV record has been populated. This Perl code looked specifically for XML files within a certain folder structure. Users could dictate the folder in which these XML files are located, allowing for limited or wide scope within a server's directory structure. This Perl code has default values for these and other input parameters (source directory, output file name and, XML source directory) if the user wishes to change them. In addition, the aforementioned batch files were created so multiple instances of this application can be executed concurrently and multiple output files can be created for multiple databases.

Perl contains modules catered to the geospatial sciences. Using the locations of the map extent in decimal degrees and the ellipsoid datum gleaned from the GIS metadata, distance and area measurements were calculated. Using the ellipsoid as an input parameter, functionality within the *Geo* module converted these geo-coordinates into quantitative distances and areas that were in the data mining and analysis process. Like the C/C++ programming language, the appropriate library must be declared at the beginning of the code using the following command:

```
use Geo::Ellipsoid;
```

This library was added onto the basic Perl installation using a simple GUI called the Perl Package Manager. This manager serves as the technical support center for Perl developers who package Perl code to perform certain tasks. After undergoing a probationary period, these libraries are accredited and offered to the Perl community

through this GUI. Various libraries can perform complex statistical analysis, matrix functions, hyperbolic functions and even derive amino acid sequences for the biological sciences. Perl has the ability to do this within the confines of its code, but these libraries access this complex code so each individual programmer does not have to do this from scratch. Instead, they make an instantiation of a Geo object and perform calculations allowed within the appropriate library such as the following:

```
my $geo = Geo::Ellipsoid->new(ellipsoid => 'WGS84', units
=> 'degrees');
```

The range function within the Geo library took in parameters to compute distances along the surface of the earth.

```
$ns_distance_mi = $geo->range($metadata{"lat1_s"},
$metadata{"Geographic_Location"}, $metadata{"lat2_s"},
$metadata{"Geographic_Location"})/ $meters_in_a_mile;
```

While most Perl code was created by the researcher for this project, this Geo module was created by Jim Gibson, a Perl developer and uploaded through CPAN, the Perl developer network by which Perl code is tested and shared (Gibson 2008). It was the intention of the programmer to derive area size from values extracted from metadata. Since it had already been done, there was no need to recreate this application.

This is an extremely short, but useful computation because it converted between decimal degrees and real world distance. Given the shape of the earth is an oblate spheroid and different ellipsoids are used with different coordinate systems, it is often problematic converting between these absolute locations. With an oblate spheroid, users

must consider both radii (semi-major and semi-minor) and the flattening ratio when making computations. Different spheroids and therefore coordinate systems on which they are based such as the Clark 1886, GRS (Geodetic Reference System) 1980 and WGS (World Geodetic Survey) 1984 use different radii and flattening ratios to represent the shape of the earth. As a result, formulae to convert between absolute location and real world distance are dependent upon these factors. Perl developers with GIS experience have noted this problem and developed the aforementioned modules to make these conversions as geodetically precise as possible. The distances and areas of the extent, measured on the surface of the earth as opposed to a Cartesian plane is one such advantage of these GIS Perl modules. While not required by the FGDC, these areas were associated with metadata quality using data mining techniques or queried by the user.

Output of Perl Processing

A sample CSV file created by this Perl process is shown in Table 3. The MART Website (http://www.uncg.edu/~tjmulroo/MART) contains a link to the raw CSV file if the user wishes to perform other processing to it. With regards to the FGDC processing, an example of the HTML output created from this Perl processing is shown in Figure 3 . For FGDC-required features, a color coding system of Green and Red dictate compliancy or non-compliancy. These colors are represented as variables and can be changed within the variable declaration portion of this code.

As shown in Table 1, there are 7 required metadata elements that need to be populated as dictated by the FGDC. In other words, all GIS data layers created for distribution in the United States need to have the following elements populated: Data Set

Title, Publication Date, Language, Data Theme, Abstract, Metadata POC and Metadata Date. Essentially these required elements serve to describe the dataset, when it was created and who filled out this information. While some dependencies do exist, such as the metadata POC having a position and organization, only population at this top-most level is required. In the same vein, the FGDC suggests population of 15 other elements. These suggested features serve as a cross-section of facets that speak to a layer's spatial, attribute and descriptive integrity. These suggested features include vertical accuracy, reference system, metadata standard name, location of data set and the responsible part of the data set.

In most instances, organizational requirements will extend these FGDC required and suggested stipulations to include facets such as source information. If data are digitized from an independent digital source such as DOQ imagery, it is important to know where the source was acquired, how the information was acquired and most importantly the time period that the raster imagery represents. These facets can be easily programmatically added into the Perl code as part of MART. All that would be required is the path to the attribute within the XML schema and an accompanying hash element to store the information.

This lag between geographic reality and its representation in the digital environment serves as pressing issue for 21st century geospatial scientists. Features such as roads are always being created. However, it is important that databases representing these features or imagery that includes these features also be updated and catalogued. Everyone from the intelligence community to those using real-time GPS for driving

directions is dependent upon this information. It is necessary to develop a cohesive means between the creation of features, their representation in the digital environment and the distribution of these representations in a timely manner.

If all FGDC-required metadata elements have not been satisfied, the number of total populated elements is listed and the missing elements are populated in the appropriate column. In the same vein, the records that are missing FGDC-suggested elements are signified as yellow. In both instances, the total number of populated metadata elements is calculated and a percentage is displayed for users and decision makers.

Conclusion

In order for a specific field residing within various XML metadata files on a server to be analyzed, it first needed to be assimilated within one file in an organized manner. This was done using the Perl programming language. The goals of Perl within this research included the following:

- Incorporated information residing in various XML metadata files and placed them into a single file
- Navigated the XML schema of these files and placed them within their appropriate tag/attribute value. This file was used in the large data analysis and data mining process
- Compiled results of FGDC compliancy and placed them into a separate HTML file for use by decision maker

The advantages of Perl in this instance are many-fold. They include the following:

- Perl is designed to extract information from text and strings
- Perl is open source and can run on almost any operating system
- The Perl user and support community is large and knowledgeable
- Perl has functionality to traverse any XML schema given the location of the tag, which is already known
- Perl has functionality to perform complex GIS computations
- Perl can concurrently open and edit files, allowing for analysis from one file to be dynamically written to another file

While the R programming language may be able to write the FGDC report files, Perl served as a better alternative because of its text and file management capabilities. While Batcheller (2008) has shown that VBA/ArcObjects can programmatically access the appropriate metadata elements and statistical software such as Excel or SPSS/SAS (for later processing) can assess these elements, this code used open source programming technologies and can run under any operating system and computing environment. In the proceeding sections, R created graphs, charts and rudimentary results from large scale data analysis and data mining. Regardless, less than 800 lines of Perl code were required to copy metadata values from an unlimited number of XML metadata files and place them into spreadsheet format while checking the existence and compliancy of these values at the same time. It would take a human many days if not weeks to perform these tedious data preparation tasks. From this format, complex analysis on the qualitative and quantitative attributes was performed.

Data Analysis

Data in tabular format serves as the formal framework from which large scale qualitative and quantitative statistical analysis can be performed. These data are stored in comma separated value (CSV) format. This CSV format is not software dependent and works cohesively with open source statistical software packages. Upon population of this CSV file from the XML metadata via Perl programming techniques, a module running the statistical analysis was automatically deployed.

The analysis of these data was performed by a software package called *R*. *R* is a statistical software package and serves as the open source counterpart to the *S/S++* suite. One advantage of *R*, like Perl, is that it is open source and therefore free and able to run on a variety of operating systems. *R* also has a macro programming language of the same name, which can batch process statistical operations that one would perform at a command line interface, in addition to employing constructs such as decision structures, loops and logical operators that one sees in a typical programming language. Like Perl, the software code is readily available and can be edited by savvy software programmers to fit their needs. *R* is able to perform large-scale data analysis on large data sets such as those derived from GIS metadata, some of which contain more than 4,000 individual elements.

Perl code was used to test compliancy to FGDC-required and suggested features displayed within the contents of a table dynamically written to an HTML file at the time of creation of a master metadata table. R could have just as easily performed this activity. However, the power of R is in its ability to perform statistical analysis on

quantitative elements of metadata and graphically render its results in graphs, charts and histograms.

Descriptive statistics such as mean, median and mode can be derived from quantitative data using R. Quantitative data such as dates and accuracies, while few and far between, exist within GIS metadata. The publication date, an FGDC-required element, can be quantitatively compared from one record to the next. Within the code, a module was written to convert FGDC date conventions (YYYYMMDD) input by the user to ratio data (YYYY.XXXX) for use in these computations. The function even checked for the presence of leap year and computed it appropriately given the author's leap year birthday. For example, February 16, 2009 would be computed as the floating point number 2009.126027. Metrics such as range, temporal mean (the average of all dates) and temporal median were calculated for these ratio data much easier than the calendarical notations.

Using an R command called read.csv, the file name passed at a command line opened up the master metadata file created using Perl. Then all data on a column by column basis from the master file wishing to be processed or analyzed was read into a data structure specific to R. In the example below, the information containing the publication date was read into R. The resulting data were much like a one-dimensional array, but has methods and classes built into them that can access statistical and graphical function. In the code seen below, the individual components of the YYYYMMDD date convention were parsed to their individual components so they can be converted to ratio data.

```
ydata$dateParse<-gsub("'", "" ,as.character(ydata$r02_Publication_Date))
ydata$NumChars<-nchar(as.character(ydata$dateParse))
ydata_good<-list(date=ydata$dateParse[ydata$NumChars == 8])
ydata_good$Month<-as.integer(substr(ydata_good$date,5,6))
ydata_good$Year<-as.integer(substr(ydata_good$date,1,4))
ydata_good$Day<-as.integer(substr(ydata_good$date,7,8))
```

Once the creation of these ratio values was done in a separate function, functions appropriately named *mean*, *median*, *var* and *sd* built within R performed some basic descriptive statistics on the dates which have been converted to ratio format.

```
temporal_mean<-mean(ydata_good$theDate)
temporal_median<-median(ydata_good$theDate)
temporal_variance<-var(ydata_good$theDate)
temporal_std<-sd(ydata_good$theDate)
```

For example the temporal mean for a data set computed in R may be 2006.1092. This number represents the average date of all publication dates for an entire data set. Since this number has little value to the end user, another function converts these numbers back to a month/day/year format (2/8/2006). This popular format is used almost exclusively in the United States as European date conventions

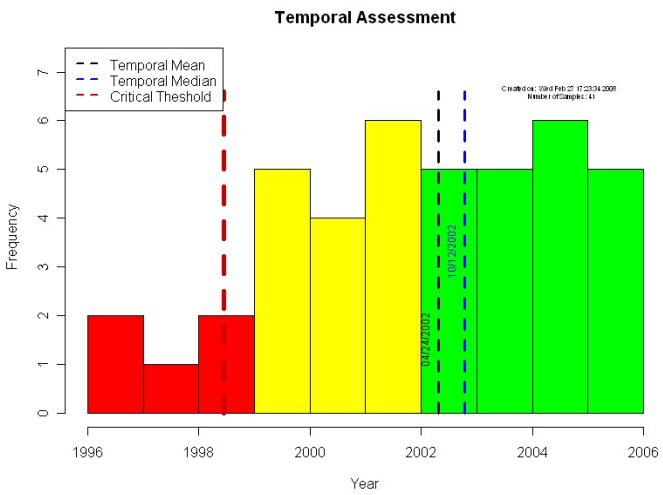

Figure 4. Sample R Output Showing Histogram of Distribution of Publication Date

use the day first. The format used in the United State is sometimes referred to as Middle Endian format, as opposed to the Big Endian format used to stored dates with GIS metadata.

In addition to computing these metrics, histograms were created from these data to graphically show the distribution of publication dates amongst all GIS records that have been populated.

Figure **4** shows an example of the histogram created from this information.

Using a command called *switch*, output was written to a JPEG file instead of a default window within R. Using a command called *hist*, a histogram diagram was created using the newly calculated dates in ratio format as an input parameter.

```
hist_x<-hist(ydata_good$theDate, plot = FALSE, breaks = numBreaks)
```

The number of breaks was calculated to determine the level of histograms based on user input data. Code to color these histograms based on a critical value (1.5σ below mean in this case) represented this temporal accuracy and its relationship to these centrality metrics. Red (layers that need immediate attention), yellow (layers that will need attention soon) and green (layers that do not need attention) were shown to represent temporal accuracy of individual layers. In addition, vertical lines to shown the location of the temporal mean and median within the histogram were drawn within the R code using the following command:

```
lines(c(temporal_median, temporal_median),c(0,max(hist_x$counts) *
1.1), lty = 2, lwd = 3, col = "BLUE")
```

While some layers by nature may have poorer temporal accuracy (1990 census data, for example) and little can be done about it, other layers such as roads, buildings and utilities require constant upkeep. This histogram gave GIS decision makers a way to visualize this temporal information and equiped them with another tool to aid in the decision making process.

Finally, using commands called *sink* and *cat*, new HTML files used to store the histograms written to a JPG image were combined with text information such as the spatial mean, spatial median, spatial variance and spatial standard deviation.

```
sink("publication.html")   # Write data to HTML file
cat("<html><head><link href='master.css' rel='stylesheet'
       type='text/css'></head><body><TABLE><TD>")
cat("<IMG SRC = 'output_temporal.jpg'>")
cat("</TD><TD><B>Temporal Mean: </B>", get_date_text(temporal_mean), "<BR>")
cat("<B>Temporal Median: </B>", get_date_text(temporal_median),"<BR>")
cat("<B>Temporal Variance: </B>", get_time_text(temporal_variance), "<BR>")
cat("<B>Temporal Standard Dev.: </B>", get_time_text(temporal_std), "<BR>")
cat("</HTML>")
sink()
```

This *cat* command dynamically wrote HTML code which can be interspersed with results from R analysis. The result is a non-static web page which changes from one execution of the code to the next captauring changes in the GIS metadata for that spatial database.

Another quantitative attribute collected from metadata that was graphed in this format is the horizontal accuracy. Horizontal accuracy speaks directly to how accurate (the position on a map versus its position in reality). Horizontal accuracy is a direct reflection of the credibility of the map, map creator and organization publishing the map. It is implausible to represent geographic phenomena with 100% accuracy, but cartographers work off of National Mapping Accuracy Standards to help guide the amount of acceptable error within a map (USGS 1947).

Figure 5. Definition of Horizontal Accuracy

For GIS metadata, horizontal accuracy can be derived from a number of different sources. When using a GPS unit, horizontal accuracy can be derived from the PDOP (Positional Differential of Position). If working with data extracted from analog sources such as paper maps like traditional USGS 1:24,000 Topographic Quadrangle Maps, the horizontal accuracy can be derived from that layer's metadata. It is important to note this accuracy because the validity of the map may come into question if it is not known. This error is scale dependent. The NMAS (National Mapping Accuracy Standards) have created the formula below which accounts for a 90% confidence interval. These equations are summarized below using the source scale denominator as the multiplier:

For Scales ***1:20,000 or greater***: .033 inches * Scale of Map

For Scales ***1:20,000 or lower*** : .02 inches * Scale of Map

Using this formula for a 1:25,000 map, the horizontal accuracy is 500 inches, or approximately 12.7 meters. All data appearing on a 1:25,000 scale map should have a horizontal accuracy of less than 12.7 meters (USGS 1947). The histogram graphs representing horizontal accuracy showed thresholds for conventional or popular mapping scales and is shown in Figure 6.

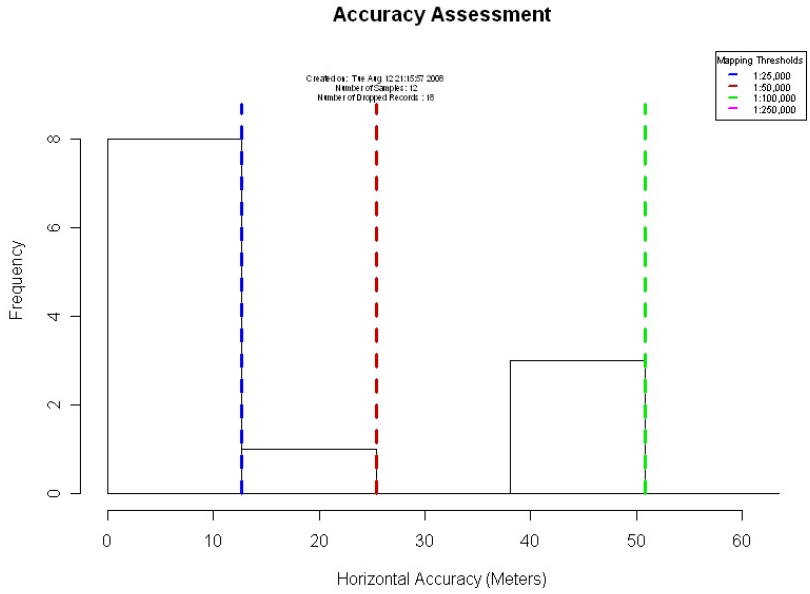

Figure 6. Sample R Output Showing Histogram Distribution of Known Spatial Accuracies from a Sample GIS Database

The histogram was created with division of 12.7 meters (1:25,000 scale map) to the acceptable horizontal accuracy for popular scales such as 1:50,000, 1:100,000 and 1:250,000. These are popular map scales employed by the USGS. Code used to draw these static lines among the dynamic histograms is below.

```
lines(c(12.7, 12.7),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3,
col = "BLUE")
lines(c(25.4, 25.4),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3,
col = "RED3")
lines(c(50.8, 50.8),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3,
col = "GREEN2")
lines(c(101.6, 101.6),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd =
3, col = "MAGENTA")
leg_accuracy = c("1:25,000", "1:50,000", "1:100,000", "1:250,000")
```

If working with data layers that lie to the right of the threshold based on the map scale you are working with, it should be noted on the map in a disclaimer statement and used judiciously.

This graph used histogram divisions based on NMAS standards, which are typically used for analog data or data that were once in analog format. This would include scanned USGS 1:24,000 imagery and DLG (Digital Line Graph) data, which are the point, line and polygon work used to compose these maps converted to digital format for use in the GIS world.

However, with the proliferation of digital data that never existed in map format that is inherently different from its analog counterpart, a new standard was devised. These standards are defined in the National Standard for Spatial Data Accuracy. The NSSDA “provides a common language for reporting accuracy to facilitate the identification of spatial data for geographic applications” (FGDC 1996, p. 3-1). The

NSSDA accounts for new technologies and more precise techniques that did not exist when NMAS was devised.

While different from NMAS because it does not define threshold accuracies, the NSSDA serves to consult agencies on helping to define internal accuracy standards based on a variety of parameters. According the NSSDA, the horizontal accuracy for a 1:24,000 scale map according to NSSDA standards would be about 13.9 meters, compared to 12.2 meters according to NMAS standards. However, NSSDA standards does this at a 95% confidence interval, compared to 90% confidence interval for NMAS (FGDC 1996). While arguments can be made about what standard is required given the source and intended use of the data, the histogram graphs shown divisions based on NMAS standards. A cross-walk between NSSDA standards can easily be done. However, given the fluidity of NSSDA because of its reliance on RMSE (Root Mean Square Error), static denominations dictated by NMAS serve as better reference points for the sake of this graph. In addition, much source data such as 1:24,000 USGS Topo Quadrangles and DOQ imagery once existed in analog format before their conversion back to digital format such as Digital Raster Graphic (DRG) and DOQ imagery. Metaphysical arguments exist about which standard is applicable for which data source, but this fall outside of the scope of this research. As a result, both standards are mentioned to serve as a guide for GIS data developers.

Other forms of accuracy do exist which can be potentially captured from GIS metadata, but their very nature makes assessment problematic at best. Attribute accuracy is the quality that the description of a feature within the confines of tabular information

matches it real world description (Chang 2002). Attribute accuracy describes the rate at which labels such as road names stored within a GIS accurately reflects the features it is supposed to represent. Given the heavy reliance on the human element to confirm this accuracy via ground truthing or some other resource intensive activity, calculating this accuracy using MART is an impossibility at this time and for the foreseeable future. Please consult the Discussion for further elaboration and other forms of accuracy.

Using statistical analysis techniques, descriptive statistics about qualitative data such as frequency histograms were created. Compliance to the FGDC standards highlighted in Table 1 was checked and compared as part of this qualitative analysis to render Boolean (True or False) results. In addition, information required for the data mining process was gleaned from quantitative data (publication date and horizontal accuracy) such as mean, median, mode, standard deviation, variance and frequency and be displayed. *R* also employed a command called *switch* so graphs and charts can be written to image formats compatible within the web environment instead of the standard output window found within the *R* operating environment. These image formats include JPG (Joint Photographic Experts Group) and PNG (Portable Networks Graphic). These images, combined with textual output can be written to HTML files and referenced once the referring pages have been dynamically edited, and later compared to the output of other databases or time periods for use in the decision making process. The code to perform these steps is shown in APPENDIX D.

Rule Development using Data Mining Techniques

Applications of data mining and KDD (Knowledge Discovery in Data) transcend many different fields. Data mining can be used for making stock market predictions, sequencing DNA, detecting fraud and determining insurance premiums. In addition, the diversified goals of data mining are dependent upon these fields. While data mining and KDD are sometimes defined as a 'non-trivial process of identifying valid, novel, useful and ultimately understandable patterns in data' (Fayad 1996), it does so in various ways. Based on data dimensionality and robustness, data miners can explore trends and make predictions, populate missing values within large databases, cluster or group data into like categories or derive business rules based on hidden relationships exhibited in the database with greater certainty.

GIS data quality can be quantitatively measured a few different ways, most notably by its temporal and horizontal accuracy. If Tobler's (1970) First Law of Geography can be applied to a chronological scale, more recent GIS data should be 'better' than older data. In addition, lower horizontal accuracy dictates a better adherence to geographic reality, thus validating analysis and maps created with these data. However, what factors lead to the creation of this better data? Data mining techniques were used to determine dependencies and create rules to determine what factors contributed most towards high quality data. Are these differences personnel related? Are these rules determined by geographic location? Data mining techniques were created that best yield these hidden relationships within the GIS metadata.

Depending upon the type of data, KDD and data mining can take many different approaches. They include Bayesian Networks, Decision Trees and Neural Networks. This study explored these different concepts to discover rules and attribute dependencies within large data sets. The goal of this study was to apply these techniques to GIS metadata to discern hidden trends within these data. Because the data collected transcends nominal, ratio and interval data, it explored attribute relevance and data generalization so a clear presentation of characterization results could be made. Lastly, it explored the significance of applying these techniques to GIS metadata. Given the human element intrinsic to GIS data, rule development techniques can be applied to this form of data with significance that can be variable in nature.

Association Rule Learning

A popular application of unsupervised data mining is using computer algorithms to look for unseen trends or groupings within large datasets. The 'beer and diapers' phenomenon is an example of this association mining. In looking at a large database of purchases, a store found that on Fridays, the purchase of diapers was highly correlated with the purchase of beer. As a result, diapers and beer were placed near each other in the store, which in turn increased purchases. No one had ever asked that question before, so it is impossible to query within the confines of a database using the supervised techniques.

While the 'beer and diapers' example may be urban legend, it highlights the importance of data mining in our every day lives. Association mining falls within a subset of unsupervised data mining techniques which rely solely on computer

applications (Klösgen and Żytkow 2002). In particular, association mining or association rule learning is the discovery of trends within large database. Germane to this research, the database is the information gleaned from GIS metadata. Association rules were applied to this GIS metadata to determine if undetected relationships did in fact exist and could be used by GIS decision makers. Is horizontal accuracy related to location? Does the GIS data created by a particular organization have poorer temporal accuracy (i.e. is older) than other organizations that have contributed to the database? Association mining can help answer these and other questions.

The Transaction Table

Given the original application of association rule learning to consumer purchases, the aptly named transaction table looks and feels differently than the spreadsheet format used to store GIS metadata build and discussed previously. A transaction table merely has only two columns: a transaction identifier and what was purchased in that transaction. A traditional example of this transaction table is shown in Table 4. In this example, a buyer made a purchase of only milk and eggs, which is signified by purchase number 367. For the next buyer, they purchased beer, chips, milk and eggs. This transaction table can get quite large and detailed.

Transaction ID	Purchase
367	Milk
367	Eggs
409	Beer
409	Chips
409	Milk
409	Eggs
897	Eggs
897	Cheese
897	Bread

Table 4. Example of Transaction Table

Given this transaction table, association rule learning algorithms can create rules. A rule is an implication to help explain user behavior. Given the transaction table, a rule

may be '{Milk} -> {Eggs}.' This means that whenever a user buys milk, they also buy eggs. Essentially these algorithms count the number of occurrences each unique item with the occurrence of other items and groups of items. Compound rules can be created such as '{Cheese, Eggs}->{Bread}.' This means that whenever cheese and eggs were bought, bread was also bought. Given the permutations of transactions which bleed into the field of combinatorics, users must be mindful when looking at large databases of these transactions and in turn interpreting its results.

While rules can be created from any permutation of occurrences, the strength of this rule is defined by its confidence. This confidence explains the percentage of times that a rule is true. In the previous {Milk} -> {Eggs} example, the confidence for this rule is 1.0, meaning that when milk is bought, eggs will also be bought 100% of the time. However, for the rule {Eggs} -> {Milk}, the confidence will be .667, meaning that when eggs are bought; milk will also be bought 2/3 of the time. Notice in transaction 897 when eggs were bought, but not milk. Other metrics of association rule learning also include the lift and conviction (Klösgen and Żytkow 2002). However, given these measures, it is difficult to discern the usefulness of a rule. Some rules will be obvious; others will be less so. Given the frequency of a rule in concert with its confidence can help users determine its usefulness.

As this is applied to GIS metadata for the sake of this research, a spreadsheet format extracted from GIS metadata must be converted to a transaction table. This was done by the researcher using the Perl programming language. When the data were being processed from separate XML files into a single CSV file, a transaction table was also

created in the process. In essence, each cell from the CSV file or XML file was treated as a single transaction. Attributes from each transaction will be compared with the next using a separate Perl function created for this purpose. 20 different qualitative and quantitative metrics were placed into this transaction file. These varibles are highlighted in Table 5.

GIS Metadata Feature	Description
Area of Layer	Areas will be placed into nominal classes called Low, Medium and High
Update Frequency	Taken Directly from Metadata
Place Keyword	Taken Directly from Metadata
Geoid	Taken Directly from Metadata
Ellipsoid	Taken Directly from Metadata
Object Count	Count will be placed into nominal classes called Low, Medium and High
Contact Organization	Taken Directly from Metadata
Metadata Organization	Taken Directly from Metadata
Contact Position	Taken Directly from Metadata
Metadata Position	Taken Directly from Metadata
Use Constraints	Taken Directly from Metadata
Access Constraints	Taken Directly from Metadata
Number of Attributes	Taken Directly from Metadata
Publication Date	Date will be placed into nominal classes called Unknown, New, Medium and Old
Data Theme	Taken Directly from Metadata
Metadata POC	Taken Directly from Metadata
Metadata Date	Date will be placed into nominal classes called Unknown, New, Medium and Old
Spatial Resolution	Date will be placed into nominal classes called Unknown, Excellent, Medium and Poor
Location	Using Latitude and Longitude gleaned from metadata, the area of the country will be quantified as Northwest, Upper_Midwest, Northeast, Southeast, Lower_Southeast and Southwest.
Metadata POC	Taken Directly from Metadata

Table 5. Different Variables Placed into the Transaction Table

The Perl code below shows an example where values are taken directly from the metadata. In this code, the transaction number, incremented within a loop to reference a

new record, is written with the contact organization and separated with a tab. Using the substitution operator 's/', all spaces are substituted with underscores. Doing this without spaces will cause less confusion for the data mining modules in Perl.

```
# Find the contact organization and replace the spaces with an underscore
       my $contact_org_trans;
       $contact_org_trans = $metadata{"n13_contactorganization"};
       $contact_org_trans =~ s/ /_/g;
       print TRANSACTION_FILE
_$transaction_number,"\t","Contact_Organization=".$contact_org_trans,"\n";
```

Given that association rule learning makes groups based on those nominal designations created above instead of quantitative models via regression or even fuzzy logic, the publication date, horizontal accuracy, area of feature, object count and number of attributes were placed into different categories using compound *if…then* statements. Categories include high, medium and low for the object count. These nominal categories are highlighted in Table 5. The example below shows how the object count is placed into different categories. The values directly reference the aforementioned hash element which acts as a type of array.

```
# Find object count and put it into one of 3 different values
if($metadata{"n11_objectcount"} <= 10)
{
print TRANSACTION_FILE $transaction_number,"\t","Object_Count=Low","\n";
}
if($metadata{"n11_objectcount"} >= 11 && $metadata{"n11_objectcount"} <= 100)
{
print TRANSACTION_FILE $transaction_number,"\t","Object_Count=Medium","\n";
}
if($metadata{"n11_objectcount"} > 100)
{
print TRANSACTION_FILE $transaction_number,"\t","Object_Count=High","\n";
}
```

The rest of this Perl code is highlighted in APPENDIX C. Figure 7 is an example of the transaction table created from GIS metadata. Each transaction or selected attribute from each record, created on a new line and then separated by a tab.

```
1       Horizontal_Accuracy=Unknown
1       Location=Northeast
1       Responsible_Party=Mike_Carmichael
2       Publication_Date=Medium
2       Data_Theme=Local_GIS_Data
2       Metadata_POC=Timothy_Mulrooney
2       Horizontal_Accuracy=Unknown
2       Location=Northeast
2       Responsible_Party=Mike_Carmichael
3       Publication_Date=Medium
3       Data_Theme=Local_GIS_Data
3       Metadata_POC=Timothy_Mulrooney
3       Horizontal_Accuracy=Unknown
3       Location=Northeast
3       Responsible_Party=Mike_Carmichael
4       Publication_Date=Medium
4       Data_Theme=Public_Safety
4       Metadata_POC=Timothy_Mulrooney
```

Figure 7. Example of Transaction Table Created from GIS Metadata

Notice the Publication Date reflects that it is Medium and that the location is in the Northeast. In addition to making rules based on the responsible party and spatial integrity of the data, rules related to location and size of the areas were also made. This is the manner in which the geographical component was applied to this research and how these geographical differences were reflected and ultimately gleaned through GIS metadata.

Making Association Rules with GIS Metadata

The application of these techniques within GIS metadata to qualitative features,

location and quantitative metrics is revolutionary. With the creation of a transaction table, rules like those created for consumer purchases were applied to GIS metadata. With the transaction table dynamically created by the researcher based on values from the GIS metadata database, results from this association rule learning may from one use to the next in the future.

A Perl module developed by a Perl developer aptly named Association Rules uses the dynamically created transaction table to create association rules. This module was developed by Dan Frankowski and made available through CPAN (Frankowski 2008). After the HTML, CSV and transaction file (in .txt format) have been created, the module to perform this association rule learning was called. The parameters for this file were:

- Transaction File – This is a tab delimited file showing transaction number (record) and a unique attribute from that transaction. This was dynamically created while writing data from separate XML files to a unified CSV file.
- Support – Number of transactions in which the item-set appears. Support level 2 will only allow an example such as '{Milk} -> {Eggs}'. Support level 3 will allow another item as part of the item or set pair.
- Confidence Threshold – Number between 0 and 1 that represents the minimum percentage of occurrences between one set and another (i.e. rule)
- Maximum frequent set size to look for (optional)

Using the example from Table 4 and this Association Mining Module created by Dan Frankowski, a set of rules can be made at a .5 confidence threshold. An example of this output is as follows:

```
2 0.667 1 1 2 Eggs => Milk
2 1.000 1 1 2 Milk => Eggs
```

This means that using Support 2, the confidence of {Eggs} -> {Milk} is .667, meaning that when eggs are bought, milk was bought 2/3 of the time. The expression '1 1 2' means that 1 item is on each side of the transaction, resulting in 2 total items. This is intuitive given the support level of 2. When milk was bought, eggs are also bought 100% of the time.

To further exemplify how the association rule learning can be applied to GIS metadata, a sample transaction table was run from 30 GIS data layers at support level 2 and confidence .75. While a much larger run will be discussed later, from these 19 variables collected and using a support level of 2, more than 450 rules can be found with a confidence greater than .75. If the support level was greater than 2, there would be many more rules to represent the compound permutations of items that could exist in a set. Some of this sample output from this run is as follows.

```
#     Con   Rule
7     1     Location=Southwest=>Metadata_Date=Old
8     0.8   Location=Upper_Midwest=>Publication_Date=Old
29    0.967 Geoid=D_WGS_1984=>Update_Frequency=As_needed
4     0.8   Metadata_Date=Unknown=>Area_of_Layer=High
11    0.8   Metadata_Date=Unknown=>Horizontal_Accuracy=Unknown
8     0.8   Contact_Organization=NC_DOT=>Metadata_Date=Unknown
5     1     Data_Theme=Wetlands=>Number_Attributes=Medium
```

Some of these rules can be extremely useful. The first rule states that all GIS data from the southwest United States will have an old metadata date. These sample GIS data were taken from a nationwide dataset and the metadata date may need to be revisited. For the next rule, all GIS data from the Upper Midwest will have an old publication date 80% of the time. This should tell GIS managers that these data in this part of the country need to be revisited to ensure that the most updated data are maintained. While this sample database consisted of 30 layers, other GIS databases may consist of hundreds or even thousands of layers. When budgeting or scheduling resources for future update efforts, it may take days for a GIS data steward to sift through thousands data layers to determine which GIS data are the oldest. Association rule learning helped to direct users to which cross-section of data needs to most immediate attention.

While the case illustrated above correlates geographic location with temporal accuracy, other cross-sections include the contact organization for a GIS data layer. Another rule stated that when the contact organization was the North Carolina Department of Transportation, the metadata date was unknown. Metadata date is an FGDC-required element and its omission will be highlighted in the output. Given that association rule learning transcends all metadata elements, these rules can be applied to these features in concert with other FGDC-related elements, locations or other interesting fields.

One negative side to association rule learning is the plethora of information created from these rules that the users must sift through. Using the example above, all layers that had a data theme of wetlands had a moderate number of attributes. That may

just be a function of the spatial data standards used to define a layer's naming conventions, attributes (number and names) and domain tables. SDS-FIE (Spatial Data Structures for Facilities, Infrastructure and Environment) is used by the government to help define a layer's name (wetlands are called wetland_area) and the name.number of attributes (for wetlands, 62 attributes are collected). In addition, according to SDS-FIE standards, the nutrient class attribute can have only one of 5 values (Eutrophic, Mesotrophic, Oligortrophic, TBD and Unknown). This is useful for interoperability to serve as a standard baseline between all government organizations which implement these data. Regardless, having a moderate number of attributes may have nothing to do with data quality and as a result, this rule may have little utility. However, a rule showing a certain organization creating layers with a low number of attributes may not be adhering to attribute standards such as those defined in SDS-FIE. In this case, the GIS manager may need to explore this issue further. This will be addressed in Chapter VI.

Because the algorithms used to define these rules do not know the difference between useful and needless information, the relevance of interestingness always arises. In using the aforementioned example of purchases from a grocery store, the purchase of salad and salad dressing are highly correlated. This is obvious, but the computer algorithm does not know that. For the GIS database that was queried, all GIS data from the state of Ohio used a WGS 1984 Ellipsoid. That probably is not useful information. Therefore, the results of this data mining, frequency of rules and their confidence should be used by the end user to determine the interestingness of its results. Studies by Geng

and Hamilton (2006) delve into this interestingness issue and is addressed in the Discussion.

Association mining or rule learning is an unsupervised data mining technique used to make associations or relationships between different attributes of a database. Association rule learning has been utilized by grocery stores to look for groups of products that are purchased with each other for the purpose of promotional placing or product placements. The application of association rule learning to GIS metadata is revolutionary in this research. With each GIS metadata record-attribute pair being treated as a transaction, variables gleaned from GIS metadata were related to each other with some degree of confidence. Results from these rules can help guide and define supervised techniques to help users fully understand the various dimensions of their complex GIS metadata database.

Exploratory Data Analysis

In addition to these unsupervised techniques, this research will allow users to define searches and perform hypothesis testing. Users can test specific hypotheses using EDA (Exploratory Data Analysis) techniques (Klösgen and Żytkow 2002). Using EDA techniques, the user will be able to interact with the data so searches on any and all variables collected can be made. Data, relationships and correlations between these 43 different variables were visualized and analyzed at a variety of different dimensions. From there, users can create hypotheses for future testing or decide the appropriate tools based on the type of data (nominal, ordinal or ratio). This helps to address the interestingness problem that critics of data miners seem to broach. User interaction

determines what is interesting, thus nipping that tautological dilemma in the proverbial bud. EDA uses applications of traditional statistical techniques to help guide rule development and knowledge development. EDA serves as a logical complement to unsupervised data mining techniques because it brings together the best of both worlds – using the user's a priori knowledge to help direct queries while also giving an unbiased or impartial look at all of the data to see if any future EDA should integrate results from these unsupervised techniques.

PHP and MySQL

Various techniques allow associations or groupings to be made within the data gleaned from metadata information. However, this can be done a variety of different ways. Unsupervised techniques use computer algorithms to create these groups or associations. Supervised techniques allow users to help guide or define the parameters for these groupings. The application of these techniques to this metadata information, the effectiveness of these techniques and ultimately the assessment of these techniques by the user community served as the intellectual impetus of this research.

Users may want to perform supervised queries on the data to look at groupings or associations that they think may be important. For example, a user may want to query all GIS data layers where the data were published before a certain date. They may also want to query data that does not have a horizontal accuracy defined. These user-defined queries are very important to GIS decision makers to help dictate future data and personnel resources.

It is essential that users have a means to define their own queries. PHP serves as the means to accomplish this digitally. PHP (Hypertext Protocol Language) is a server side programming language that allows users to interact between HTML form elements such as drop-down menus and text boxes and the database (Ullman 2008). It is presumptuous to expect users to interact with the raw data within the confines of a complex database. PHP allows users to do this using an environment with which they are familiar. In this day and age, the web serves as this perfect environment.

Like R and Perl, PHP is an open source programming language. While server side scripting languages such as Cold Fusion and ASP do exist, they can be expensive (Cold Fusion) or catered to a particular software package (ASP). PHP allows users to publish pages using the extension 'PHP' instead of more traditional extensions such as HTML, HTM, MHT or TXT that are suitable for display in the web environment.

In essence, PHP code is passed to a server and is interpreted into HTML code which can be displayed by any web browser. PHP is only installed on the server computer, therefore sidestepping security issues and compliance issues that one may see with client side applications such as Java and JavaScript. PHP has advantages over traditional web formats such as HTML because PHP is not static and can be changed based on user input. In addition, PHP was designed to interact with databases and render results of queries passed in by users. Since PHP action is server side, it can not be seen by the end user and input parameters such as passwords, database names and their absolute locations can be saved within the confines of PHP code without being seen by end users.

Information about thousands of metadata files are saved within the confines of a CSV file created using Perl. While PHP can query and manage this non-proprietary spreadsheet format, PHP interacts much easier with a MySQL dataset. MySQL is a lightweight, flexible and free database that can manage relatively small data sets (Ullman 2008). Compared to other large data configurations and relational database management systems, metadata information collected in the course of this project with 4 dozen attributes and thousands of records would classify as relatively small and manageable within the confines of MySQL and PHP.

Querying the MySQL Database Using PHP

In this research, a web page was created using PHP that allows users to query the fields that are highlighted in Table 1. This web page is shown in Figure 8.

User-Defined Exploratory Data Analysis

This form allows users to explore dimensions of their GIS metadata. Data collected transcend various data scales (nominal and ratio). Users can make compound queries of the data. Select and populate all of the appropriate fields that you wish to query. Leaving all fields blank will return all records.

Required Elements

Data Set Title:	NONE	*Search Pattern*	*
Publication Date:	After:	Before:	NOT FOUND ☐
Language:			
Data Theme:	*		NOT FOUND ☐
Abstract:	*		NOT FOUND ☐
Metadata POC:	NONE		
Metadata Date:	After:	Before:	NOT FOUND ☐

FGDC-Suggested Elements:

Spatial Resolution:	Greater Than:	Less Than:	NOT FOUND ☐
Distribution Format:	NONE		
Additional Spatial Information:	*		NOT FOUND ☐
Spatial Representation:	NONE		
Reference System:	NONE		
Lineage Statement:	*		NOT FOUND ☐
Online Resource:	*		NOT FOUND ☐
Metadata Field:	*		NOT FOUND ☐
Metadata Standard:	NONE		
Metadata Version:	NONE		
Metadata Language:	*		
Metadata Character:	NONE		
Location of Data:	Latitude (+/- DD):	Longitude (+/- DD):	
	BUFFER:		NOT FOUND ☐
Responsible Party:	NONE		
Character Set:	NONE		

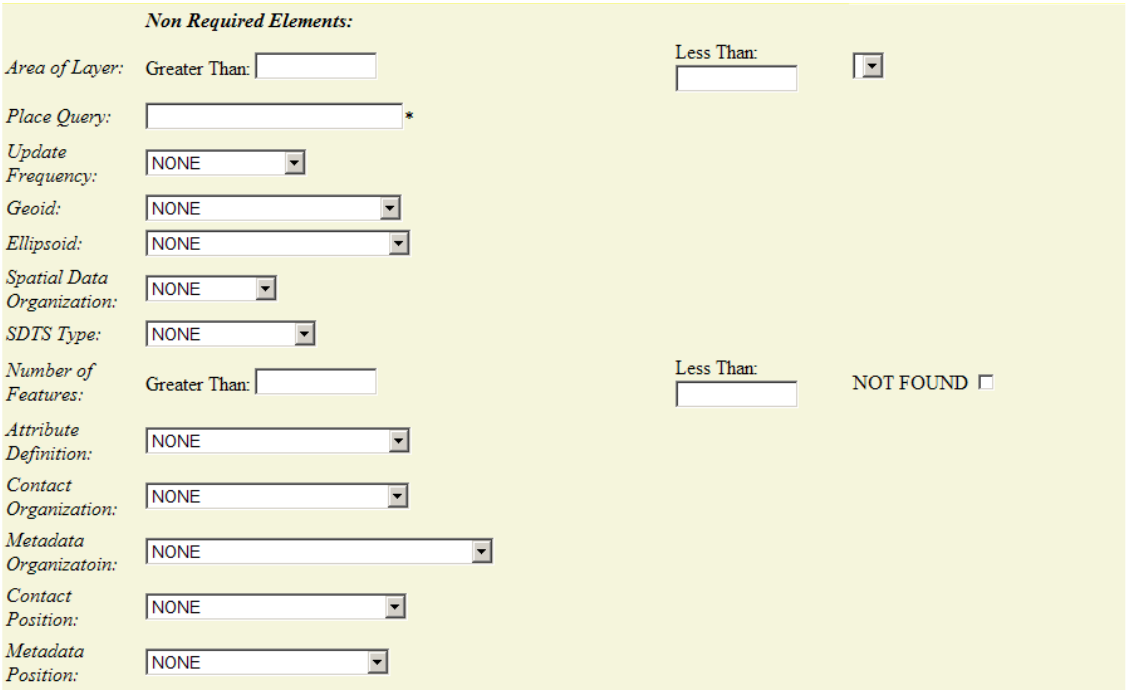

Figure 8. Example of Web Page Created Using PHP

Users can query the MySQL database that contains metadata information for all GIS data layer using three different form elements (text field, drop-down menu and checkbox). Notice the custom drop-down menu created from only those unique values in that particular attribute. Users can select all, some or none of elements to query. These queries serve as the foundation of supervised techniques and exploratory data analysis. About 40 different attributes can be queried. Only a few are shown here.

Users could enter free text for some fields or use a drop-down menu for domain specific selections. Keyword matching was implemented for free text entry to allow greater flexibility for certain elements. This keyword matching allowed users to put in partial names to allow for spelling or interpretational variations. For example, a user may want to query the process steps for all instances of the expression "sde" to show where the ESRI's Spatial Data Engine (SDE) has been used to process the data. However, some users may refer to it as "ArcSDE." In both cases, keyword matching allows hits to be found in both scenarios.

The drop-down menu form elements are a combination of static HTML code and queries on certain attributes of the MySQL database. While the text box elements were easy to code, the drop-down menus were more difficult. These free text entries worked just as well as a drop-down menu to query data; however, drop-down menus work better for domain specific data. SQL (Structured Query Language) commands must be used in concert with the newly populated attributes of the MySQL database. SQL is a stand along programming language or set of protocols that can used within a programming language designed to retrieve and manage data within a RDBMS (Relational Database Management System) (Ullman 2008). SQL is sometimes thought of as a stand alone programming language and was originally designed as such. In other cases, SQL statements are built using simple GUI elements. Within the confines of GIS software, the 'Select By Attributes' dialog (**Figure 9**) allows users to build an SQL statement from various fields for a GIS data layer. The interactive buttons allow users to make both numerical comparison using standard numerical

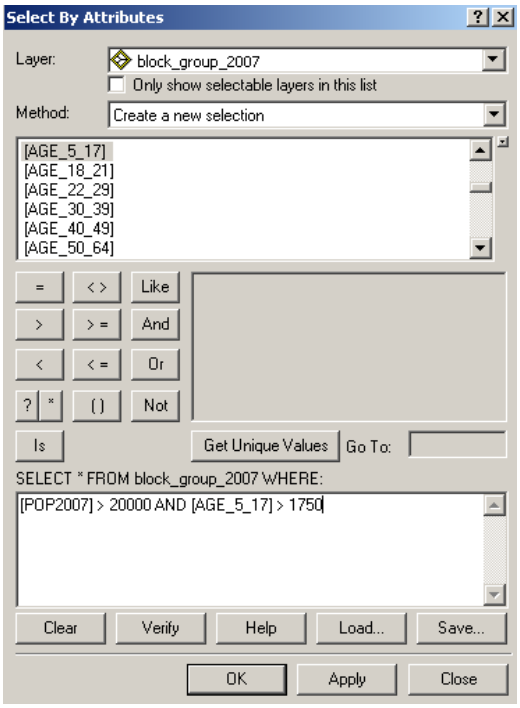

Figure 9. Example of SQL Builder in ESRI ArcMap

operators such as > (greater than), < (less than) or = (equal to), logical operators such as AND, NOT and OR to make compound queries and finally string comparisons such as LIKE or IS. As one can see from **Figure 9**, as the user interacts with the GUI elements, an SQL statement that the GIS software will understand is built.

When the user executes this statement, the features and accompanying records that satisfy the SQL statement will be highlighted on the map. Whether a casual GIS user knows this or not, anyone who has ever used this dialog has actually built and executed an SQL statement.

A command called SELECT DISTINCT in SQL selected only those unique values for a particular attribute or column in a RDBMS schema. In HTML, a form element called a SELECT created a simple drop-down menu with various possible entries without repetition.

These entries were defined by OPTION parameters nested within the SELECT element. The number and values of these entries were dependent upon the values within that particular attribute. For example, the Metadata Position field may have hundreds of different entries, but only a few unique entries such as “GIS Specialist”, “GIS Manager” and “GIS Analyst”. This SQL statement and subsequent population of the form elements compressed the drop-down menu into only those unique elements which will satisfy the query. This customized form creation can not be done using strict HTML coding techniques. In addition, another large dataset may have an entirely different set of unique entries for the same attribute. As a result, one set of form elements for one data set are

not transferable to the same form element for another data set. An example of these custom-made form elements can be seen in Figure 8.

This supervised data mining technique was actually composed of two separate web applications. This first page loaded the page using a five step process. These steps included:

1. The CVS file created from gleaned metadata information was dynamically imported in a MySQL database. A new database instance was created as all old versions of the database were dropped.
2. Opened the MySQL database using parameters passed by the user and writes HTML code using the echo command.
3. Created table to display data and form to capture information that were passed to an actions page.
4. Created static form elements such as text boxes that will use keyword matching to make queries and check boxes to represent if an element has a value of NOT FOUND.
5. Created dynamic form elements such as drop-down menus by querying the database and selecting distinct elements of the appropriate attribute.

By doing this, the web page will be populated with elements relevant to GIS data in question within a framework that will be familiar with persistent users of the database. After the user created the criteria that will be used to search by typing in values, checking a box or selecting a value from a drop-down menu, the user will select the submit button, eliciting activity from the second page. The steps involved in this process included:

1. Values from the form elements in the previous PHP page were passed to variables in this new PHP page. These values represented the queries that wish to be performed on the database.
2. If these elements had values (i.e. the user has created a search criteria for that variable), an SQL statement querying that element in the MySQL database will be created.
3. This SQL statement were concatenated with other SQL statements created from other search criteria if they have been entered by the user.
4. This compound SQL statement executed the query of the database in one line of PHP code.
5. The records that satisfied this query will be displayed using a simple *for….next* loop using static HTML code to representing the attribute headings.

Using the dynamic form created in Step 3, users can specify criteria among the 43 different features via a combination of drop down boxes, free text and check boxes. For example, suppose the user wants to query all GIS data that satisfies the following criteria:

1. Publication was date between January 1, 2001 and January 1, 2006
2. Had an Unknown Spatial Resolution
3. Contained between 10 and 75 attributes

This may be useful because a GIS manager may want to find all spatially incomplete data created during a certain time. Given all of the information collected in the data collection process, users are only limited by their imagination. Given this scenario, the PHP code built a SQL statement. If a form element was populated, the base SQL

statement was concatenated with the populated elements. If a form element was not populated, the PHP code skips over that criteria and continues to the next one in trying to build the compound SQL statement. Given the criteria described above, the SQL statement created is as follows:

```
SELECT * from eda_table WHERE tableID > 0 AND
r02_Publication_Date > 20010101 AND r02_Publication_Date <
20060101 AND s00_Spatial_Resolution = 'unknown' AND
n23_numattributes > 10 AND n23_numattributes < 75
```

In addition, keyword matching was encoded with the confines of an SQL statement. Using the process step query as an example, if the user wished to find all instances of the string of 'sde' in the process step, the user would've just typed in 'sde' in the appropriate text box. However, behind the scenes, the SQL statement that is generated and executed would be as follows:

```
SELECT * from eda_table WHERE tableID > 0 AND
s05_Lineage_Statement LIKE '%sde%'
```

As stated before, the LIKE statement serves as the logical comparator for strings. The % symbol serves as a replacement for zero or more characters. This symbol is sometimes referred to as a wildcard. Since this symbol both precedes and proceeds the query string, any string containing the phrase 'sde' will return a match. In this way will all occurrences of the phrase 'sde' will return a match. As one can imagine, the SQL statement can be very complex depending upon user input from the form elements. Upon

execution of the SQL statement within the PHP code, all records satisfying this query were returned using a single statement such as the one seen below:

```
$result = mysql_db_query($dbname, $query)
```

Because there are various variables that can be queried via the form elements, it takes a multitude of code to build these statements. The result of this query can be saved using the database name and query string generated using the code. Using a simple loop, the results can be displayed within the confines of an HTML table. An example of these results can be seen in Figure 10

File Name	Data Set Title	Publication Date	Language	Data Theme	Abstract	Metadata POC	Metadata Date	Spatial Resolution	Distribution Format	Additional Spatial Information	Re
ammunition_storage_area.xml	Ammunition Storage Area Marseilles Training Center	20070518	English	Military Operations	The area where ordnance or other explosive/hazardous devices are stored.	Amy Howard	20070619	unknown	Downloadable Data	Vertical accuracy is not applicable to this dataset	ve da
buildings.xml	Buildings Ravenna TLS	20061127	English	Buildings	An existing structure that was created by man for occupation storage or to facilitate an activity.	Greg Edmonds	20070208	12.2	Downloadable Data	Vertical accuracy is not applicable to this dataset.	ve da
cantonment_area.xml	Cantonment Area Marseilles Training Center	20070518	English	Cadastre	This coverage represents the cantonment area for Marseilles Training Center IL.	Amy Howard	20070619	unknown	Downloadable Data	Vertical accuracy is not applicable to this dataset	ve da
carto_training_area.xml	Training Area (Cartographic) Marseilles Training Center	NOT FOUND	English	Military Operations	Cartographic representation of any area where military training is conducted.	Amy Howard	20070619	unknown	Downloadable Data	Vertical accuracy is not applicable to this dataset	ve da
					A permanently					This feature	

Figure 10. Sample Output from Exploratory Data Analysis

This output satisfies a query that was created in the interface shown in Figure 8. All attributes collected are displayed within the confines of the output table. Because of the width of the output, only a few of these attributes are shown in this sample.

The code used to create the form elements and execute the SQL statement can be found in APPENDIX E and APPENDIX F.

While PHP has been used to query databases in the past, these supervised techniques served as a means to give control to the user in order to visualize various dimensions within their geospatial infrastructure. This control was not only important from a business standpoint, but also a theoretical standpoint because of its application. Little research has been done regarding the effectiveness of these techniques to GIS metadata. In addition, viewing important elements of once disparate pieces of data within the confines of one database and interface tool captured within this research is revolutionary to the field of GIS metadata. The techniques and processes described above can be catered to popular software packages so future generations of GIS users can better assess and quantify their metadata with little time or effort.

The supervised techniques used in this research involve two basic applications. In this first one, PHP was used to populate a MySQL database from raw data gleaned from multiple XML files and created a customized web interface so users can perform complex queries. The second application executed and rendered these compound results based upon user defined criteria. PHP is a very powerful programming language. At its most basic level, it can simply write HTML code. However, database programmers can use it to interact with open source databases such as MySQL created solely for this purpose. The supervised techniques created for this research demonstrated the ease with which this can be done to query information about GIS metadata within the web environment.

Data Output

Once rules have been determined from the multitude of data and graphs representing descriptive statistics have been created in the appropriate format, they were written to a web page or another suitable output. Functions within R called *sink* and *cat* can write raw HTML code along with the corresponding variables and graphs within a web page. This code can be seen in APPENDIX D. HTML tags surrounded by the '< >' qualifiers were known. However, these values are not. Techniques interspersed these static tags with those that are dependent on the output. The dynamic creation of static and non-static features using these techniques gives traditional web authoring languages more flexibility than its predecessors. These graphs and textual output were organized amongst 3 different web pages. The creation of various output pages was done for the sake of data organization.

Another form of output was the creation of an XML standard with the sole purpose of storing output from these analyses. Because of its flexibility, XML can be catered towards a developer's need. There are literally hundreds of XML standards in existence, ranging from MathML to FicML, a markup language used to store information about works of fiction. Geographers may be familiar a variation of XML called GML (Geography Markup Language). GML is used to describe geographic phenomena and encode this information using the tag-value couplets that XML is famous for. An added feature to this research was the creation of a standard XML format so these assessments can be stored and later shared across the entire user community.

XML is an effective format to store metadata because of its flexibility. As long as it adheres to a few very loose rules, anyone can create an XML standard. While the ISO and FGDC own the proverbial market on GIS metadata standards for desktop applications, anyone can extend these existing standards. However, the attributes and values for a new standard may not match with conventional metadata editors. In addition, XML even has it own query language, appropriately named ***X***ML ***Q***uery ***L***anguage (.xql). XQL is specifically designed to query collections of XML data. While related to SQL, it is as flexible as the XML data it is intended to query.

The United States Census Bureau Factfinder Web Site (http://factfinder.census.gov) uses XQL as an easier way to query its extensive database containing data at various levels of aggregation collected by the United State Census Bureau. Instead of using web form elements such as drop down menus on multiple web pages to select all areas (census block groups, for example) and tables (median household income and average family size, for example), users can load an XQL file into Factfinder to retrieve the data. For large areas that may encompass hundreds of enumeration units, users can only select them a few at a time within Factfinder's web form elements. This selection process can be long and error prone.

The XQL file required by Factfinder needs 1) the form name (Short Form 3, for example) 2) the Table Number (H53, for example) and 3) a list of enumeration units the user wishes to query. Putting these parameters into the attribute-tagged couplet compatible with the Census Bureau can bypass that selection process and query the necessary data. This can especially be useful if users will be querying information from

the same area throughout the course of a project. While the enumeration units would be the same (called geographies in XQL), the table name would merely need to be changed in order to create an entirely new query. With the future development of MART, users may be able to query the MARTO-XML files using XQL-like convention.

An example of this output created specifically for this research is shown in Figure 11.

```
<?xml version="1.0" encoding="utf-8"?>
<Metadata>
 <MARTOContactName>Timothy Mulrooney</MARTOContactName>
 <MARTOContactEmail>tjmulroo@uncg.edu</MARTOContactEmail>
 <MARTOPhone>336-617-6868</MARTOPhone>
 <databaseName>Test Data for MART</databaseName>
 <numberofLayers>890</numberofLayers>
 <DateofAsessment>20090604</DateofAsessment>
 <FGDCCompliancy>
  <RequiredNumberofLayersFullyCompliant>628</RequiredNumberofLayersFullyCompliant>
  <RequiredPercentofLayersFullyCompliant>70.56</RequiredPercentofLayersFullyCompliant>
  <RequiredFeaturesPopulated>5583</RequiredFeaturesPopulated>
  <RequiredFeaturesPossible>6230</RequiredFeaturesPossible>
  <PercentFeaturesRequired>89.61</PercentFeaturesRequired>
  <SuggestedFeaturesPopulated>4805</SuggestedFeaturesPopulated>
  <SuggestedFeaturesPossible>13350</SuggestedFeaturesPossible>
  <PercentFeaturesSuggested>35.99</PercentFeaturesSuggested>
  <SuggestedNumberofLayersFullyCompliant>16</SuggestedNumberofLayersFullyCompliant>
  <SuggestedPercentofLayersFullyCompliant>1.80</SuggestedPercentofLayersFullyCompliant>
 </FGDCCompliancy>
 <TemporalAccuracy>
  <CritialThreshold>20051127</CritialThreshold>
  <TemporalSTD>1.96</TemporalSTD>
  <GraphLink>temporal.jpg</GraphLink>
  <NumberofMissing>103</NumberofMissing>
  <TemporalMedian>20050405</TemporalMedian>
  <TemporalMean>20050201</TemporalMean>
  <TemporalRange>9.99</TemporalRange>
  <TemporalMaximum>20071228</TemporalMaximum>
  <TemporalMinimum>19870101</TemporalMinimum>
 </TemporalAccuracy>
 <HorizontalAcccuracy>
  <HAMaximum>125</HAMaximum>
  <HAMissing>283</HAMissing>
  <HAGraphLink>horizontal.jpg</HAGraphLink>
  <HAMinimum>1.26</HAMinimum>
 </HorizontalAcccuracy>
 <AssociationRules>
  <DateofRules>20080604</DateofRules>
 </AssociationRules>
</Metadata>
```

Figure 11. MARTO-XML Standard for Output from One Data Run

This rudimentary MARTO-XML (Metadata Assessment and Reporting Tool Output) standard was divided into 5 major categories. They are 1) General Information About the Assessment, 2) FGDC Compliancy Assessment 3) Temporal Accuracy Assessment 4) Horizontal Accuracy Assessment and 5) Results from Data Mining. This MARTO-XML schema shows a minimum number of tags. Tags that qualify temporal mean and temporal accuracy units (years vs. months, for example) may be required if the complexity and variation of the data increases. For now, these units were designated in an FDD (Functional Description Document) for this standard. Nonetheless, these tags within this standard can be used and accessed by web pages or GUI applications such as those that currently interact with GIS metadata. In addition, these XML files can be saved and referenced at a later date further.

Data Archival

A feature of this application and dissertation is the ability to archive old data, replacing it with the newly created information derived from the processes described above. However, deleting these data altogether would prevent ancillary analysis such as multi-temporal change detection that may be useful in long term planning. While most of this information was stored within the confines of a new XML file, figures and diagrams referenced within the XML file must not be lost. Within the confines of the larger web site, this old information was written to another location and a referring document updated so these old data are saved in an organized manner. The copying of this information was performed via batch files. For the Windows computing environment, DOS (Disk Operating System) commands merely created a new folder, moved the

current data to an archived folder, ensuring that naming conventions with the referring documents are retained. These commands are systems operating dependent, but parallel each other.

Data Organization

Once the data were physically archived, the referring document to this newly archived information needs to be created. The *archive_toc.html* was updated by adding a link to this archival data with a Perl program. It searched the HTML code containing the link to the latest archive and write a new link using a naming convention below this code.

Testing

One downside of using open source programming techniques is that application specific Dynamically Linked Libraries (DLL) can not be created using proprietary languages such as Visual Basic. However, separate Perl programs such as those used to perform the processing tasks previously described.

A sample application of MART was run using a database of 890 individual data layers taken from the author's place of work. The data transcended all themes, data creators and accuracies. Using a batch file, the necessary commands were all placed into a single file. Upon running this batch file or scheduling it to run when activity is low, it created descriptive output in the form of the aforementioned graphs, tables and charts. These tables were stand-alone HTML pages that can be viewed via the Internet while the charts and graphs are embedded within an HTML file for viewing through a web browser. In addition, a MySQL database and web page to access was created so users

could perform query the database using supervised techniques. Finally, an association rule mining application was run to perform unsupervised techniques using this batch file.

The Test Environment

MART was run on a relatively small GIS dataset of 890 data layers transcending various themes on May 23, 2009. MART was set up in a test environment that would simulate the working conditions for a practical application of MART. Given the dynamic web display component, in this case MART was run on a web server with Internet Information Services 6 (IIS 6) in the Windows 2003 server computer. The GIS data and metadata were connected using a networked drive and appropriately referenced to the Perl and R commands modules using a batch file. This batch file can be run at any time. Upon application of this batch file, older web pages and images were renamed and archived as they are replaced with the newly created histograms and metrics. This solution is one of many that could be done to link the different applications. Virtual directories from IIS are another solution within the Windows operating system. An open source Apache web server can do the same thing as IIS.

The operating system for the test environment is a Windows Server 2003. The server only has about 135G of hard drive space, but is connected to 3 other server computers using static IP addresses. This increases the available space to just under 950G. Servers of this size would be perfect to accommodate imagery and raster data, which typically require more space than their vector counterparts. This server computer has 2.40 GHz CPU and 2.0 G of RAM.

Given these criteria, all open source packages were able to efficiently run. Perl

(Version 5.10) was able to run on this computer, as well as Perl Package Manager (PPM), which allows users to interactively add, remove and manage modules catered for a specific purpose. The PPM was used to upload GIS functionality in Perl, as well as the Association Rule Mining (ARM) module which was used to create association rules. Version 2.9 of R was able to run on this server. In terms of the web component which serves as the interface to perform supervised techniques for this research, PHP (Version 5.2.9) was used with a MySQL database. This PHP code interfaces with a MySQL database client, Version 5.1.11.

Testing FGDC Compliancy

The MART application took approximately 12 minutes to run, partly because the hardware in the testing environment was slightly older. Perl was used to assimilate the data from disparate XML files into a single spreadsheet format and then create tables and HTML output regarding FGDC metadata standard compliancy. R was used to created graphs to describe quantifiable (horizontal and temporal) accuracies within the data. Applying this metadata information to data mining techniques can also be done via open source. An existing Perl application was performed on the metadata information to discern hidden trends regarding the metadata information using user-defined levels of tolerance. If a user chooses to look for trends themselves, a PHP program was used to create a web interface which allowed users to query the database and perform supervised classification on the data.

Besides the existing ARM (Association Rule Mining) module and short GIS-related Perl functions which later proved inconsequential, all of the applications were

designed from scratch by the author. In conjunction with ARM, custom Perl code was created by the author to build a transaction table. A traditional spreadsheet format is composed of records and attributes, where their intersection forms a cell. In a transaction table, each cell is treated as a separate record. Given a simple 50 x 20 spreadsheet matrix, there are 20 attributes used to describe 50 individual records. Decomposed to a transaction table, there will be 1,000 records composed of a unique record identifier and the unique cell value for that particular record and attribute. The code to perform this decomposition was entirely coded by the author. For this sample application of MART, 890 records and 43 attributes, a transaction table composed of 38,270 records was created during the processing of MART.

In addition to the nature of the data driving decisions via unsupervised techniques, existing FGDC metadata standards drive a large part of the assessment portion of MART. A table appearing like Table 3 was created for this application. While it would be rather overwhelming to show this 890 record table, it showed and color codes each individual layer, its compliancy to FGDC required elements, compliancy to FGDC suggested elements and any missing required or suggested metadata elements. During this process, it tabulated these numbers for this entire database for an overall metadata compliancy statistic over the entire database. Figure 12 shows the results.

More than 19,000 individual metadata elements were checked against existing FGDC standards in a matter of minutes. It would take a human many days to do this. From this output of the database tested, it appears that a majority (70.19%) of the data layers' metadata adhere to minimum FGDC standards where almost 90% of these

required elements are populated. However, less than 2% of all layers adhere to the FGDC suggested metadata requirements, with 36.12% of these suggested elements being adequately populated. It appears that this test database needs some serious work.

628 out of 890 layers (70.56%) had all of the FGDC Required metadata components
5583 out of 6230 individual FGDC required elements (89.61%) were adequately populated

16 out of 890 layers (1.80%) had all of the FGDC Suggested metadata components
4805 out of 13350 individual FGDC required elements (35.99%) were adequately populated

Figure 12. Sample Output from FGDC Compliancy Report for 890 Data Layers

Displaying Descriptive Statistics

Other descriptive analysis included histograms showing the temporal and horizontal accuracy. These were created using the R programming language. A sample output from this temporal analysis for 890 layers can be seen in Figure 13. Of these 890 layers, 103 of them did not have a publication date and were excluded from further analysis.

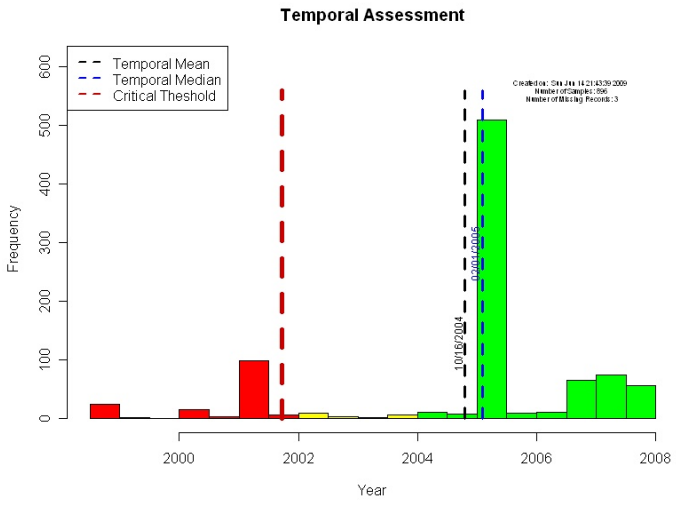

Figure 13. Output of Temporal Accuracy From Sample Application of MART

Centrality metrics performed on the publication date showed that the data show a temporal mean of 4/5/2005 and a temporal median of 2/1/2005. These metrics can be compared to dates for other databases. If another database has a temporal mean from 2006 or 2007, resources can be allocated or dedicated to update the data for this database. For the test, most of the data were published during the first quarter of 2005. In addition, a critical threshold highlights the number of GIS data layers whose temporal accuracy is more than 1.5 standard deviations below the mean. Within these red areas there exists GIS data from as early as 1998 and should appropriately be addressed were possible.

These graphs can be compared with other histograms to visualize the makeup of temporal accuracy across databases. In addition, these graphs can help guide supervised techniques. The specific GIS data layers from 1998 can queried from the large GIS database so efforts can be directed towards update these data. This graphical input can be used by non-GIS conversant personnel to give a general impression of the geospatial assets for a company and further aids in the process of knowledge discovery with an ultimate goal of decision within GIS. Information about the horizontal accuracy for the dataset can be seen in Figure 14 and grouped according to their adherence to the National Mapping Accuracy Standards for various scales of maps such as 1:50,000, 1:100,000 and 1:250,000 that are produced with some regularity.

These GIS data have horizontal accuracies that range from 1.26 meters to almost 124.6 meters, which equates to the horizontal accuracy (according to NMAS) for analog maps at a 1:250,000 scale.

However, 283 of these layers did not even have a horizontal accuracy adequately populated. Horizontal accuracy speaks directly to map quality. This important facet of map accuracy can not be ascertained for almost 1/3 of the data layers. This will give GIS decision makers insight about where to direct or re-direct metadata efforts and future data development efforts.

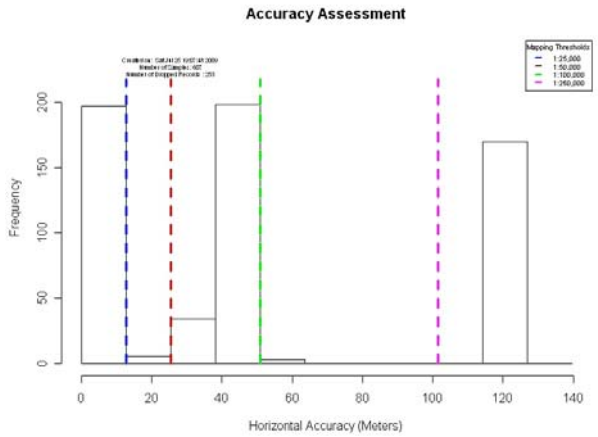

Figure 14. Output of Horizontal Accuracy From Sample Application of MART

Unsupervised Techniques on the Test Data

Unsupervised techniques were also performed on these data. This was done using the ARM (Association Rule Mining) Perl program developed by Dan Frankowski. The ARM program essentially does two things. Primarily, it creates rules. Essentially rules answer the question "When events occur, how often do other events occur?" The rules are composed of sets which can be tested with various levels of support. Support represents the total number of transactions in which the item-set appears (Frankowski 2008). Unsupervised techniques were performed on the transaction file representing the

metadata for the 890 GIS data layers. Given the permutuation and amount of results, some of these are summarized below.

At support level 1, there were 6 occurrences where the data theme is hydrology and 154 occurrences where the data contact is not found. In instances where quantitative data are used, basic divisions are used to designate them as low, medium or high, as in the case of the area of the layer shown above. There were 11 occurrences where the horizontal accuracy was medium, but 156 cases where the horizontal accuracy was excellent. In all, there were 371 of these 1-sets.

A 2-set counts the number of unique times where two of these 1-sets occur. There are 7 occurrences of a 2-set that states "Data_Theme=boundary Number_Attributes=Low." At support level 2, there were 7 occurrences where the location was the Southwest and the publication date was new. Given a 2-set is a permutation of the various 1-sets, their numbers increase exponentially and therefore quickly. There are 7,329 of these 2-sets. Using the ARM tool, an increasing number of sets can be derived from ARM: there were 15938 of these 3-sets and 77,330 4-sets.

The true power of association rule mining is turning these sets into frequent sets and association rules using a confidence metric. Confidence merely represents the ratio or probability that a rule will be true. Using the example run for test purposes, it was found if GIS data was from the Northeast part of the United States, then it would be temporally classified as 'new' with a confidence of .706. Conversely, of all metadata that was classified as new, it was from the Northeast part of the United States 97.6% of the time. These confidence intervals tell us that most of the newer GIS data for our database

is from the Northeast part of the United States. Therefore, if we were looking to allocate resources for data development to replace or update older GIS data based on date alone, the Northeast part of the United States would not be where one would begin.

There were 6,204 association rules created for support level 1 and confidence of .7 in this test. In this example all of the rules are created with 1 LHS (left handed set) and 1 right handed set (RHS). The LHS causes the RHS in this rule using the symbology LHS => RHS. In some cases the LHS can be thought of as the independent variable versus the dependent variable. Regardless, as one increases in cardinality, there can be multiple components for the LHS and RHS. However as a tradeoff, run time and computer resources would be expended to compute all of these different permutations. Regardless, in a sample run of 19,000 records in the transaction table, the execution of this program was not very computer intensive and performed at this cursory level for the sake of demonstration. Compound rules can be created can be used at the data steward's discretion.

On problem of unsupervised classification is the creation of superfluous and unnecessary rules. In addition, rules that have a high confidence, but few occurrences may not adequately portray trends for large databases. The rule below had a confidence of 1.0, but only appeared one time because DLG as a theme appeared only once. Therefore any rules related to this theme will have a confidence of 1.0

```
1 1.000 1 1 2 Data_Theme=DLG => Responsible_Party=NOT_FOUND
```

In another case, all layers with a place key of Alamance did not have a Geoid. It is likely that data for these layers were not projected, which is unacceptable for high-quality GIS data. There were only two layers that had this place key and it is important to know that this important feature used to represent absolute location does not exist. This information is invaluable and would take hours to delve from the plethora of data.

```
2 1.000 1 1 2 Place_Key=Alamance => Geoid=NOT_FOUND
```

Because of this, users must sort through both the number of occurrences within the dataset and confidence to determine rule strength. The term *support* refers to the percentage of transactions that contain a specific rule. Rules need both high support and confidence to be considered statistically significant (Klösgen, Willi and J. Żytkow 2002).

Of the 6,204 rules at support level 2, a few of these appear below. The number furthest to the left is the number of occurrences of each rule, the number next to it is the confidence.

```
67 1.000 Place_Key=North_Carolina => Geoid=NOT_FOUND
508 0.992 Place_Key=Forsyth_County => Publication_Date=Medium
353 1.000 Data_Theme=Public_Safety => Ellipsoid=NOT_FOUND
14 0.824 Data_Theme=Wetlands => Publication_Date=Unknown
```

From this, one can see that 82.4% of all wetlands do not have a known publication date. For all data layers created for Forsyth County, the publication date is medium. GIS data stewards for these layers may want to update these data in the next year or two, but not immediately. Finally, 100% of all data from the state of North Carolina and 100% of

Public Safety data do not have a known Geoid and Ellipsoid. These GIS data do not have the appropriate projection information defined. This information needs to be remedied immediately.

Other levels of support can be computed and become increasingly complex as different permutations of items and sets comprise rules. For the 890 data layers, there were 526,850 rules at support level 4. Below is a rule that has a support of 4 and confidence of .7

```
46 0.836 Metadata_POC=Timothy_Mulrooney
Update_Frequency=As_needed => Location=Northeast
Metadata_Date=Unknown
```

It says that if Timothy Mulrooney is the metadata POC and the update frequency is classified as "As Needed", then the location will be the Northeast and the Metadata date will be unknown 83.6% of the time. What does really mean? How is this information going to help me make better decisions in the future? While the computational power to create these rules is increasingly impressive, the ultimate goal of knowledge extraction becomes a complex process. It is important to realize that less may be more when interpreting the results of unsupervised rule learning.

Unsupervised techniques allowed users to extract unseen trends from GIS metadata. This is especially useful when working with hundreds or even thousands of GIS data layers. However, unsupervised techniques such as association rule mining do have their drawbacks. Of the 6,204 rules created, 4 useful ones are listed above. Hundreds of other rules are absolutely useless. Part of this has to do with the variables used in this research (perhaps too many for association rule mining) while the other has

to do with the indiscriminate nature of unsupervised techniques. In Chapter VI, scenarios are run on much fewer variables with favorable results. Regardless, the application of these unsupervised techniques to GIS data is unprecedented and served as the theoretical impetus of this research. From a practical standpoint, GIS data stewards familiar with the GIS data can use this information to better understand their GIS databases. However, while limitations exist, unsupervised techniques can be pooled with supervised techniques to create an effective and powerful knowledge discovery tool that allows for authoritative visualization of GIS metadata.

Supervised Techniques on the Test Data

The problem with unsupervised techniques is that the algorithm selects rules. Besides guiding the learning process via a support level and confidence interval, the algorithm arbitrarily creates rules with little regard to the rule's actual utility in a practical work environment. While this can be circumvented with close monitoring (see Discussion), ultimate control of the decision making process is passed from the user to the computer via these unsupervised techniques. Supervised techniques allow users to make their own queries. These queries combined with the results of unsupervised techniques can be extremely powerful tools. While these supervised queries have existed for a relatively long time, their application to metadata and execution within the real-time web environment is new.

As previously discussed, these supervised techniques use the open source programming language PHP to access the metadata elements derived from each individual XML metadata file and create a form so users can query each element. Users

can see the number of times that the metadata contact was not populated or find the layers that were published between 2004 and 2006. The opportunities are limitless for the types and variety of queries that can be performed on these data.

Users used these static and dynamic form elements such as Figure 15 to create queries

Figure 15 of the data. A number of different queries were run on the 890 GIS data layers. Users can query the required, suggested or non-suggested metadata elements. Non-suggested metadata elements are elements that are not required by current FGDC metadata standards, but may be deemed interesting or important through this research. In addition, compound queries transcending these different categories can be performed. Upon performing a query, each record or GIS data layer that satisfies the query and total number of records is displayed within a web page. A list of the queries and their results are shown in Table 6.

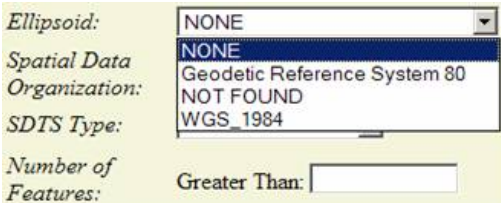

Figure 15. Example of Dynamically Created Form Element

	Query of 890 GIS Data Layers	Number of Records Satisfying Query
FGDC Required Features	How many layers have the word 'water' in the title	5
	How many layers where published between Jan 1, 2004 and December 31, 2006	433
	How many layers do not have a publication date?	103
	How many layers do not have a metadata date defined?	217
	How many layers had a spatial resolution of greater than 50 meters?	12
	How many layers had no lineage statements?	289
	How many layers did not have a responsible party adequately populated?	93
	How many GIS data layers did not have the metadata standard populated?	304
Non-Suggested Elements	How many of the layers encompassed an area between 10 and 20 square miles?	5
	How many layers did not have an SDTS type defined?	28
	How many of the layers did not have a contact organization defined?	152
	How many of the layers did not have a contact position defined?	145
	How many of the GIS data layers had the contact position listed as 'GIS Analyst'?	43
Compound Queries	How many of the layers were published before January 1, 2004 and did not have a spatial resolution defined?	100
	How many of the layers had all of the following missing: metadata date, spatial resolution and publication date	34

Table 6. Sample Queries Run Using Supervised Techniques

Conclusion on Test Application of MART

MART seamlessly ran within the confines of a typical information infrastructure. Using open source programming languages, meaningful metrics and displays can clearly articulate the health of a large GIS database. Data mining techniques allow users to view various dimensions of data. While the output from unsupervised techniques are displayed among the output, users queried the collection of metadata information housed in a MySQL database and accessed using a web interface. Information from a relatively

small database of 890 data layers has shown the utility of MART as a means to assess data fitness and guide the decision making process.

In this test application, while most of the GIS data layers had the FGDC required metadata elements populated, much less of the FGDC suggested metadata elements were populated. This analysis shows which layers and which specific metadata elements are missing. With a little more programming work, agency specific standards above and beyond these FGDC standards can be calculated and displayed. For this data layer, the temporal accuracy of the data appears to be clustered within a one year time frame. However, some data goes back more than 10 years. In addition, almost 1/3 of all data layers are missing horizontal accuracy.

Using supervised techniques, GIS managers can see that 289 data layers have no lineage statements. Using unsupervised techniques, GIS data related to wetlands does not have a metadata date, and FGDC suggested element, defined. While it does not say why, these issues need to be addressed. In addition, GIS-specific Perl commands were used to calculate the areas of extent and distances on the surface of the earth. This was done to see if relationships between these quantitative measures intrinsic to the data layer had any relationship to metadata quality. Perhaps layers that were covered smaller areas had better horizontal accuracy. Few association rules of this nature with high confidence were derived from the data. Using supervised techniques, users were able to query layers covering high areas. Little relationship between this coverage area and metadata quality can be found. As stated previously, relationship between location and certain themes were derived. However, this was done using information directly derived from the GIS

metadata. While useful as a descriptive measure, Perl modules used to make GIS calculations fall outside of the scope of metadata assessment and are not an effective indicator of metadata quality. Output from this analysis is stored in a standard XML file for future reference or change detection analysis on a regular basis.

From this information, users can make better decisions as to where, how and when to allocated resources within their geospatial enterprise. GIS managers can tell which facets of metadata and the data themselves to tackle for the test database.

CHAPTER IV

RESULTS

While the underlying technologies used to perform these analyses would be of interest to programmers and developers, the intended audience for an application of this nature is the data steward, data manager or GIO (Geospatial Information Officer) for an agency or organization. This person has intimate knowledge of the GIS data and is ultimately responsible for any exports or deliverables of these data dispensed to outside agencies. This person reports directly to GIS or IT management and decision makers so resources such as hardware, software, personnel and subcontractor services can be allocated for upcoming fiscal periods. MART helps to guide these high level decisions by quickly assessing information about these data at variety of dimensions.

Acceptance Within the Computing Community

Judging a computer application's acceptance by the technical community hinges on a number of different factors. First and foremost, it must be able to do its intended purpose, but other more extraneous factors elicit users' intrinsic wants, desires and ways that they comprehend things. Since these factors vary from user to user and testing environment to testing environment, a general statistic to judge its acceptability by the computer users must be implemented. A qualitative test was needed to see how well this application is liked, for the lack of a better word.

Assessing the effectiveness of a particular software application such as MART can be done a variety of different ways and can often be problematic given the scale, scope and usership of this application. From a strict quantitative standpoint, a metric known colloquially as a McCabe Constant measures the cyclomatic complexity of a structured computer program. It measures the complexity of a computer program via linear dependencies as one performs step-wise progression through a program's decision stuctures, function calling and subseqent return statements. Programs can range from simple to complex and ultimately unstable (McCabe 1976). The McCabe constant is used frequently to test program complexity with propietrary languages such as C/C++, Visual Basis and even Java. While Java is free to download, Java technically can not be considered open source because it runs on a propietrary software plathform. Perl has recently developed a McCabe constant module to assess program complexity within the open source environment.

In addition, an application called Doxygen serves as a source code documentation generator tool. While documenting popular languages such as C/C++, Java and Python, it also creates class hierarchy diagrams so users visualize relationships and connectivity between different modules of code. While a Doxygen filter has also been created for PHP and Perl, one does not currently exist for R.

However, little interface between the end user and MART is required. Modules run via open source techniques can be sequentially scheduled to run using a scheduling application that comes with most operating systems. In Microsoft Windows, this is the appropriately named Scheduler application. In UNIX systems, this can be done via a

cron job. The end users of this application are intended to be GIS managers and analysts as opposed to developers and programmers. As a result, code efficiency and integrity is secondary to the effectiveness and utility of this application. Therefore, a metric to measure these more cursory and cosmetic characteristics of this application is required.

Since the large scale integration of technology into society, researchers have worked on paradigms to assess its effectiveness and potential acceptance by the computing society. What may work for one group of people many not necessarily work for another group. Essentially researchers want to answer the question "What causes users to accept or reject a technology?" The testing and assessment portion of this research will focus on techniques to help ask and ultimately answer that question.

It is difficult to quantify the abstraction of innovation and its adoption within a larger community. Rogers and Shoemaker (1971, p. 219) used the term "complexity" to describe "the degree to which an innovation is perceived as relatively difficult to understand and use." Later, Tornatzky and Klein (1982) found that compatibility, relative advantage, and complexity serve as underlying and effective indicators across the spectrum of innovation types. Finally, Davis (1989) proposed a TAM (Technology Acceptance Model) discussed previously, which established two primary determinants to answer the fundamental question of what causes people to accept or reject technology. These factors were 1) Perceived Usefulness and 2) Perceived Ease of Use.

Perceived Usefulness is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance" (Davis 1989, p. 320). This Perceived Usefulness factor elicits some of the following questions:

- Will MART make my job easier?
- Does MART enhance the analysis of GIS data and metadata?
- Does using MART save me time?
- Does using MART allow me to accomplish more work than would otherwise be possible without it?
- Does MART increase my productivity?
- Is MART useful in my job?

Perceived Ease of Use, however, refers to the "the degree to which a person believes that using a particular system would be free of effort" (Davis 1989, p. 320).

- Is MART easy to use?
- Is MART easy to interact with?
- Does interacting with MART require a lot of mental effort?
- Do I have trouble understanding what MART is trying to tell me?
- Are the results produced by MART understandable and coherent?

TAM can be applied to a number of different technologies, ranging from Global Positioning System and in-car navigation units to multimedia entertainment and wireless communication. TAM can sometimes be thought of as the manifestation or logical evolution of the Mean Opinion Score (MOS). The MOS has been used for more than 50 years by the telecommunications industry. It is used to rate the quality of speech samples among communications lines. Subjects are usually asked a few questions about the voice quality of potential communications lines. Listeners use a rating scale with 5 different levels and accompanying values. They are as follows: Excellent (5), Good (4), Fair (3),

Poor (2) and Bad (1) (White 2002). The MOS is the average, or mean of all of these individual scores. In order for a communications or telephone system to be implemented, it must achieve a minimum MOS score. For example, for some companies, a communications system must achieve a minimum score of 4.0 in order to be deemed toll quality and be subsequently released to the public. Otherwise, it must undergo modifications until it can achieve this minimum score (Sevcik 2002). While TAM does tie together different facets of end user acceptance using complex quantitative methods, the basis for these methods are grounded upon subjective opinions like those seen in MOS decades earlier.

For each of these different scenarios, questions assessing the effectiveness, ease of use, attitude and behavioral intention of use vary. Researching each of these different applications, something such as MART would best fit under e-learning with regards for a testing environment. While there are many definitions of e-learning, one of them is the "learning facilitated and supported through the utilization of information and communication technologies" (Jenkins and Hanson 2003, p. 14). These e-learning Information and Communication Technologies (ICT) utilize the Internet as a vehicle to disseminate information to as many geographically diverse people as possible. It is within this e-learning paradigm that information can be shared, knowledge acquired and ultimately applied in the working world. MART parallels these ideals for the GIS profession.

TAM research has been used by websites to test cognitive and operational dimensions in which trust is gained in e-commerce applications. This is important for a

variety of theoretical and practical reasons. While making an e-commerce purchase, TAM research can determine where potential buyers go off track and the point at which incentives may be required to incite this person into making a purchase (Gefen et al. 2003). This application will integrate principles of TAM with the MOS to determine how trust and acceptance is garnered in the GIS user community via a transparent process for assessing metadata in a timely and efficient manner.

The adoption of these technologies essentially works at two levels – the individual and the organization. For an individual, the focus is on the acceptance of MART (Dasgupta, Granger & Mcgarry, 2002). The acceptance decision and ultimately the intention to use MART serves as the cornerstone for this testing environment and an examination of the factors affecting a GIS professional's decision to accept MART is the basis of this testing.

The Null Hypothesis

Using the research model proposed by Masron (2007) for e-learning environments, a slightly reduced TAM model was used to assess the behavioral intention within the context of MART.

As a result, the research hypotheses based on this model were as follows:

H1: Perceived Ease of Use for MART has a significant effect on the Perceived Usefulness of MART.
H2: Perceived Ease of Use for MART has a significant effect on the Attitude Towards Using MART
H3: Perceived Usefulness of MART has a significant effect on the Attitude Towards Using MART.
H4: Perceived Usefulness of MART has a significant effect on Intention to Use MART.
H5: Attitude towards using MART has a significant effect on Intention to Use MART.

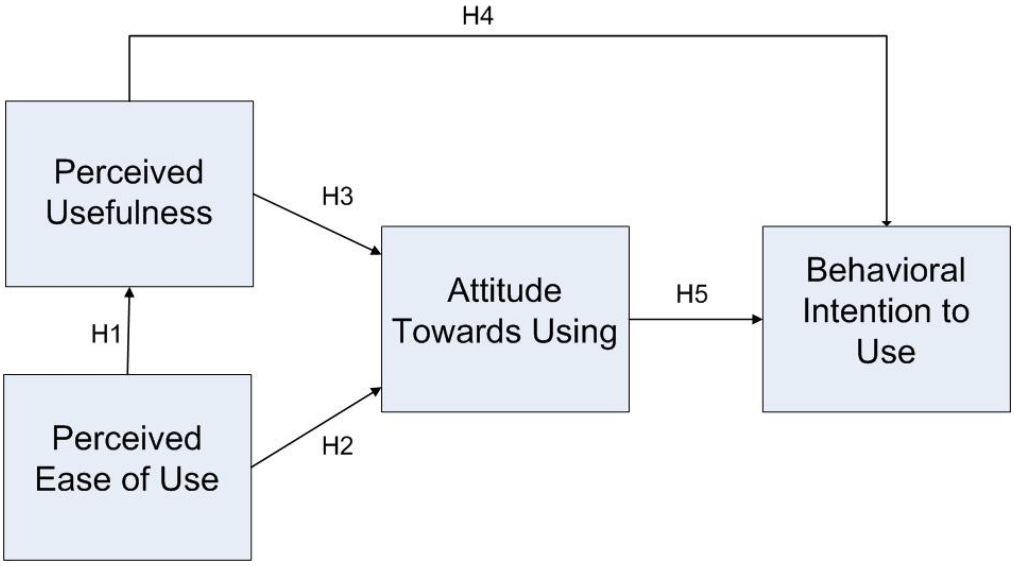

Figure 16. MART Research Model (Masron 2007)

The TAM Questionnaire

While questionnaires can take on a variety of different formats, most literature on TAM refers to the 7 point Likert scale. The Likert scale uses ordered responses on a bipolar measurement scale to assess the level of agreement or disagreement with a statement. Some scales do have an even number of responses (4, for example), which force respondents to choose one side of the mean or the other. Rensis Likert originally devised a 5 point Likert scale whose values include "Strongly Disagree", "Disagree", "Neither Agree nor Disagree", "Agree" and finally "Strongly Agree." While 10 point Likert scales are used in research, the 7 point scale is a reconciliation between these two scales and has also been used in TAM research before (Meyers et al. 2005). These questions are highlighted in Table 7.

Perceived Ease of Use (PEOU)

EASE1: I found MART easy to use.
EASE2: Learning to use MART would be easy for me.
EASE3: My interaction with MART was clear and understandable.
EASE4: It would be easy for me to find information at the MART web site.

Perceived Usefulness (PU)

USE1 : Using MART would enhance my effectiveness in learning.
USE2 : Using MART would improve my work performance.
USE3 : Using MART would increase my productivity at work.
USE4 : I found MART useful.

Attitude Toward Using (ATTITUDE)

ATT1: *I dislike the idea of using MART.*
ATT2: I have a generally favorable attitude toward using MART.
ATT3: I believe it is (would be) a good idea to use MART for my work.
ATT4: *Using MART is a foolish idea.*
Note: Reversed item.

Intention to Use (ITU)

INT1: I intend to use MART in the next 3 months.
INT2: I will return to MART often.
INT3: I intent to visit the MART web site frequently for my job.

Table 7. Measurement Items and Individual Questions Used in TAM

In addition to this assessment for the application of TAM, descriptive information collected about 40 respondents were also collected. This descriptive information anonymously asked users about their age, title, experience with GIS and experience in the facets of GIS germane to MART and metadata assessment. This experience was expressed in the approximate number of hours per week that the respondent works on GIS data development and GIS metadata. Lastly, users were asked to enter free text about their impressions of MART and advantages or disadvantages of MART. A questionnaire incorporating the Likert scale elements from Table 7 to be assessed in TAM along with descriptive elements was created in the web environment. Like the entire MART application, it was posted on web server. Once the respondent has

interacted with the MART application, they completed the entire assessment. Using server side programming technologies, a routine was written to write the results to a database. Once all of the assessments are complete, TAM analysis of the data can be done out of this database. This information is shown in Figure 17. It asks 26 questions of the respondent. The first 15 questions ask speak to the effectiveness of the application as it relates to TAM. The last 11 ask for information about the respondent that can be used in other analyses.

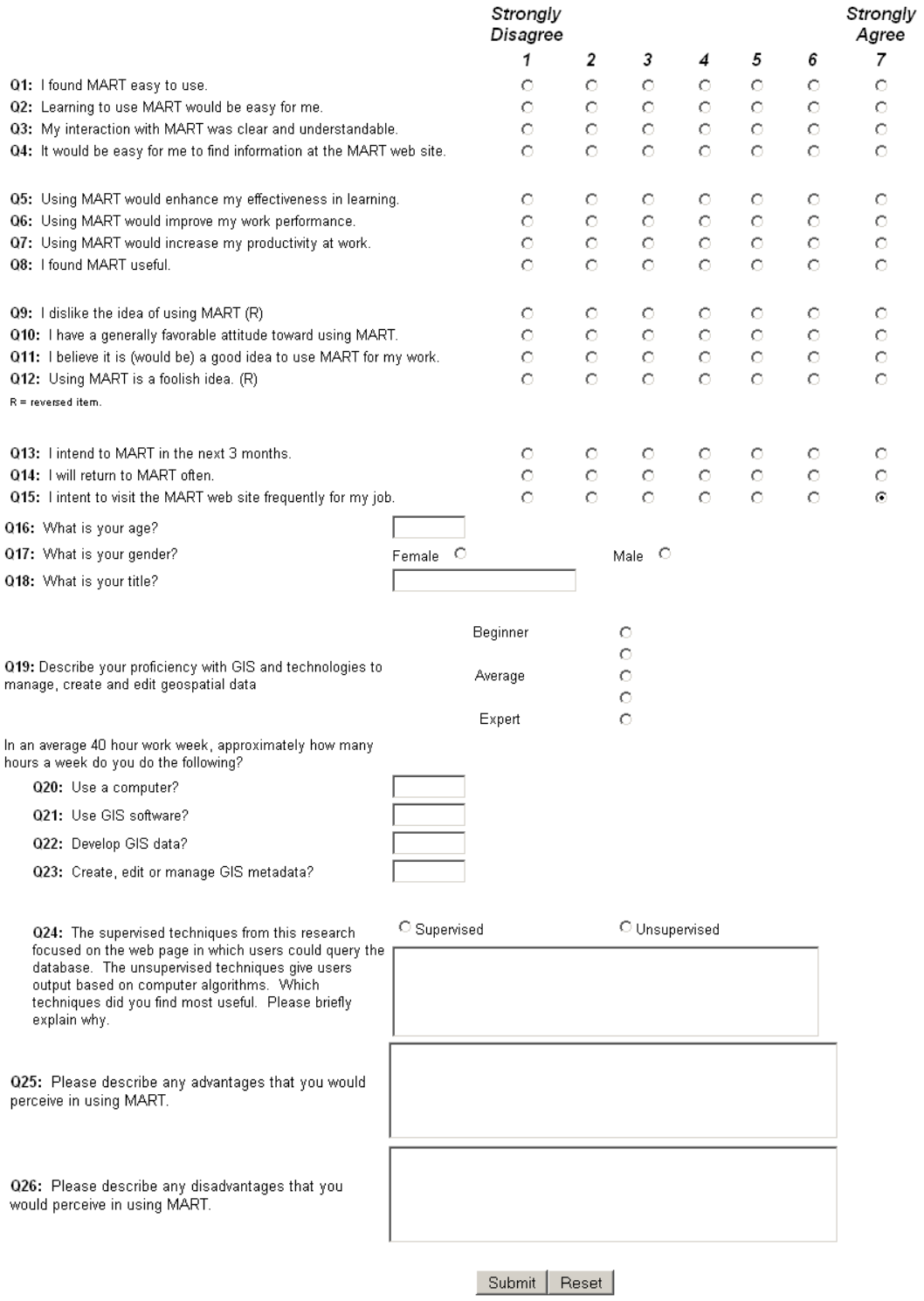

Figure 17. Web Form Used to Collect Information from Respondents for TAM

Testing Methodology

The testing of MART was done between 6/1/2009 and 7/10/2009. 40 GIS professionals, transcending all ability levels and educational backgrounds were asked to assess the output of the metadata assessment, analysis, supervised techniques and unsupervised clustering results. All testing was done in the digital environment. MART was run on a test server. This test server replicates a typical working environment, with networked access to the test GIS data and the appropriate permissions.

One sample application of MART was run and its results were displayed within a framed web page which acts as a table of contents. Users are able to view this table of contents which show output from various forms of output, clicking on elements they wish to see in the larger frame. This table of contents is shown in Figure 18. Users were asked to interact with these web pages, graphs and output files at their own speed and make their own formulations about the information presented and its utility to a geospatial enterprise. Users could navigate the MART output as many times or in whatever fashion as they wished. Using another web page, users were asked their opinion about the output of MART using the questions shown in Figure 17. There were no time constraints on either web page.

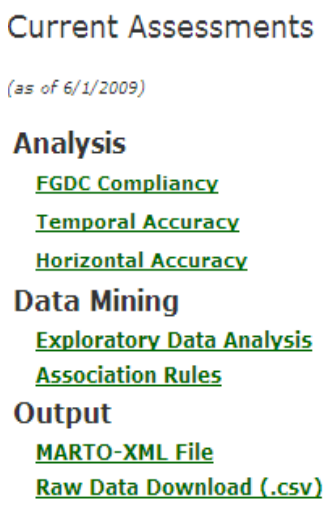

Figure 18. Table of Contents Used to Show Output of MART

Opinions from these professionals were solicited from the author of this work based on his working experience with the subjects. The author has more than 10 years of experience working in academic, industry and government GIS. In addition, while the author works in the field of GIS science and perpetuates advances in the field of GIS technology, many colleagues work GIS as a tool to quantify their research and their experience with GIS is tangential at best. Regardless, all of the people completing the survey have experience with GIS, albeit their proficiency and approach to it varies from user to user.

Users were asked to fill out the form from Figure 17. While Davis' (1989) original study contained 28 questions to assess the *Perceived Usefulness* and *Perceived Ease of Use* components, there were only 15 questions related to a user's opinion of MART via the Likert scale. 8 questions of the questionnaire solicit information about the user while the last 3 responses allow the user to enter free text regarding their opinion of the favorite of the two data mining techniques and advantages and disadvantages of MART. Using server-side programming technologies, answers from these results were recorded in a database and processed using the techniques described previously.

Testing Results

Table 14 in APPENDIX A shows the raw output from respondents taking the survey based on their opinion of MART. Some precursory information was collected to get information about respondents, their experience with GIS and more specifically metadata to see if TAM could be modeled more precisely if need be. Using SPSS software (Version 15), some basic descriptive statistics can be used to describe the

demographic composition of respondents. The 40 respondents were composed of 19 women and 21 men. Their ages range from 23 to 54, with a median age of 35, an average age is 36.03 and a standard deviation of 9.18 years. Of the 40 respondents, self-described titles ranged from GIS Analysts and Educators to Planning Technicians, Statisticians and Engineering Technicians.

In terms of GIS and computer experience, respondents described themselves as being above average in terms of GIS competence. Out of a 40 hour work week, they said they average more than 32 hours a week on the computer and use GIS software for almost 19 hours per week. On average, they spend almost spend 6.5 hours a week working on GIS data development, but less than ½ an hour a week on GIS metadata. These results are highlighted in Table 8. These respondents represent an adequate cross-section of the GIS user community from academia, the public sector and private enterprises. While some of these users work solely in GIS data development and the creation of GIS data for maps and analysis, others have a peripheral role in GIS, either as a manager, lab administrator or planner. Regardless of this experience, most disconcerting is the fact that these GIS respondents spend 18 times more time developing GIS data than working on the GIS metadata used to describe this information that will be

	Mean	High	Low	Standard Deviation
Use a Computer	32.44	36	25	5.4
Use GIS Software	18.82	30	4	8.21
Develop GIS Data	6.49	25	0	6.06
Work on GIS Metadata	.346	2	0	.527

Table 8. Given a 40 Hour Work Week, 40 Respondents Were Asked How Often they Perform Activities

used long after they are gone. Using the 7-point Likert scale, users were asked to answer the questions from Table 7. They ranged from 1 (Disagree) through 7 (Agree). The results are in posted in Table 9. In conjunction with a t-test used to compare the means for each group of questions, an upper and lower bound, in addition to a mean, were calculated at a 95% confidence interval. As shown by the means and bounding limits, respondents reacted most positively towards the ease of use in using MART. This is probably due to the presentation of results in the web environment, which is familiar with all users. Little interaction with the output is required. Users responded most negatively towards the intention to use. While results were still not negative, actual implementation of MART to cater to their GIS seemed to be the largest stumbling block. Based on user responses, this may be due to unfamiliarity with open source applications, compatibility with the current operating system, determining ways in which to implement this application on their server computers or perhaps the limited size and setup of their geospatial infrastructure.

Advanced Analysis

While TAM uses statistical techniques as its foundation, it extents further beyond those described previously. In addition, TAM looks to quantitatively assess the potential effectiveness of a tool based on innate characteristics of MART. TAM is based the relationship between an application's Ease of Use (questions 1 – 4), Perceived Usefulness (questions 5 – 8), Attitude Towards using MART (question 9 – 12), Behavioral Intent of Use (question 13 – 15) from Table 7.

Question		Median Score	Average Score	Mean at 95% CI Lower	Upper
EASE1	Perceived Ease of Use	6	5.750	5.3096	5.8654
EASE2		6	5.575		
EASE3		6	5.700		
EASE4		6	5.325		
USE1	Perceived Usefulness	5	5.575	5.110	5.702
USE2		5	5.250		
USE3		5	5.400		
USE4		5	5.400		
ATT1	Attitude Toward Using	5	5.175	4.7940	5.2560
ATT2		5	4.850		
ATT3		5	4.875		
ATT4		5	5.200		
INT1	Intention to Use	4	4.275	3.6638	4.1696
INT2		4	3.625		
INT3		4	3.850		

Table 9. Responses of Individual Questions from 40 Respondents

TAM can be done a variety of different ways to help determine a technology's acceptability within the digital environment. Using MART as a paradigm for an e-learning model, a variety of null hypotheses are proposed to determine the usefulness, ease of use and ultimately the utility of MART within a GIS infrastructure. This was previously done by Masron (2007) in order to implement information and communication technologies in college classrooms, but has been updated to account for MART and the intended use and audience of MART. Using various qualitative dimensions as criteria, ultimately the acceptability of MART can be determined using TAM analysis. As previously stated, TAM looks to find relationships between the Perceived Ease of Use of MART, Perceived Usefulness or MART, Attitude Towards Using MART and the Intention to Use MART. 40 respondents answered 15 questions pertaining to these components. The answers to these questions are summarized in Table 9.

Acceptability within the user community ultimately relies on the research hypotheses based on the various components of the TAM research model proposed in

Figure **16**. Other models highlighting slightly different relationships could be proposed, but this was used because it had been implemented previously in e-learning environments. However, before analysis is done, reliability statistics were computed to measure the internal consistency of the data.

This is done using a Chronbach's alpha instrument. Chronbach's alpha is computed using the number of respondents in the set, the variance of the data and mean of the covariance between all members of the set. While there is no universal threshold to determine data consistency, Hair et al. (1998) suggested a minimum threshold between .6 and .7. As per Table 10, only 1 of these values (Perceived Ease of Use) is between .6 and .7 while two of the values (Perceived Usefulness and Attitude Towards Using) are between .7 and .8. The Chronbach's Alpha constant for the Intention to Use component is .807, which is considered excellent (Nunnally 1978). Given these values, it can be surmised that the questions posed for the respondent serve as a reliable quantitative tool and should be treated as such.

Metric	Chronbach's α
Perceived Ease of Use (PEOU)	.659
Perceived Usefulness (PU)	.763
Attitude Towards Using (ATT)	.770
Intention to Use (ITU)	.807

Table 10. Chronbach's Alpha Used to Measure Reliability

To help understand the individual factors that contribute to any potential inconsistency, principal component analysis was performed on each of the individual questions to help determine their potential contribution to the variability of the observed results. Four factors were calculated, based on the different components of the research hypotheses to be tested.

Question	PEOU	PU	ATT	ITU
EASE1	.814			
EASE2	.813			
EASE3	.797			
EASE4	.620			
USE3		.796		
USE1		.795		
USE4		.778		
USE2		.480		
ATT4			.667	
ATT1			.647	
ATT2			.572	
ATT3			.539	
INT1				.731
INT3				.496
INT2				.480
% of Variance Explained	56.325	11.932	9.242	7.037
Cumulative Variance	56.325	68.256	77.499	84.536

Table 11. Principal Components for Rotated Factors

After rotation, the Perceived Ease of Use accounted for 56.33% of the variance. The Perceived Usefulness components account for 11.93%, Attitude Towards Using accounted for 9.24% while the Intention to Use factor accounted for 7.04%. Table 11 shows the items and factor loadings for the individual factors. Finally, some other basic correlations were run between potentially dependent factors such as age and sex to help determine their potential contribution to the results. However, no significant correlation was found between participants' age, gender and even self-described GIS experience versus dependent variables such as Perceived Ease of

use, Perceived Usefulness, Attitude and Intention to Use that will be used in the TAM analysis.

After some precursory components analysis was performed to help explain both the reliability and variance of the factors on the final results, TAM analysis was performed. In terms of the statistical and mathematical theory, TAM analysis is not revolutionary in creating innovative and complex metrics to explore various dimensions of data. What TAM does is create new ways in which compartmentalized data gleaned from user input can be compared to each other compartmentalized data using simple regression analysis. Answers regarding respondents' opinions of MART transcending a variety of different attitudes are merely regressed against each other. These statistical relationships between different groups of questions (PEOU, USE, ATTITUDE and ITU) serve as the basis of the research hypotheses to be tested.

Individual linear regression analyses were conducted based on 40 surveys collected from the study. In testing Hypothesis 1, a regression analysis was performed using Perceived Ease of Use as an independent variable and Perceived Usefulness as the dependent variable. The results from this analysis are seen below.

Perceived Ease of Use vs. Perceived Usefulness (H1)

Independent Variable	β	Standard Error of β	T	P	R^2
Perceived Ease of Use	.611	.115	5.295	< .001	.409

Later both components highlighted in H2 and H3 were tested against the attitude towards using MART for another regression model. These results are highlighted below.

Perceived Ease of Use and Perceived Usefulness vs. Attitude Towards Using (H2 & H3)

Independent Variable	β	Standard Error of β	T	p	R^2
Perceived Ease of Use	.513	.147	3.772	< .01	.657
Perceived Usefulness	.046	.127	.363	> .05	

Finally, both components highlighted in H4 and H5 were tested against the behavioral intention to use for one last regression model. These results are highlighted below.

Perceived Usefulness and Attitude Towards Using vs. Behavioral Intention Towards Using (H4 & H5)

Independent Variable	β	Standard Error of β	T	p	R^2
Perceived Usefulness	.293	.143	2.073	< .05	.427
Attitude Towards Using	.233	.1183	1.271	> .05	

Testing Conclusions

Table 12 and Figure 19 graphically show the output from the regression analysis used to test the research hypotheses shown in Figure 16 and calculated above using linear regression modeling using SPSS. From a first glance, the results are quite promising. At a 95% confidence interval, three of the research hypotheses are supported. In H3 (Perceived Usefulness of MART has a significant effect on attitude towards using

MART), Perceived Usefulness has very little effect on the Attitude Towards Using MART using the linear regression model.

However, given the strong correlation between Perceived Ease of Use on Perceived Usefulness as indicated by the high t value, Perceived Ease of Use is a large contributing factor towards the Attitude Towards Using MART component and has much more of an effect on Attitude Towards Using MART than the Perceived Usefulness. The web presentation of output and the presentation of graphics probably play into the Perceived Ease of Use components.

Hypothesis	Relationship Tested	Results
H1	Perceived Ease of Use for MART has a significant effect on the Perceived Usefulness of MART.	Supported ($p < .001$)
H2	Perceived Ease of Use for MART has a significant effect on attitude towards using. MART	Supported ($p < .01$)
H3	Perceived Usefulness of MART has a significant effect on attitude towards using MART.	Not Supported ($p > .05$)
H4	Perceived Usefulness of MART has a significant effect on intention to use MART.	Supported ($p < .05$)
H5	Attitude towards using MART has a significant effect on intention to use MART.	Not Supported ($p > .05$)

Table 12. Summary of Research Hypotheses

However, a Perceived Usefulness component (USE1: Using MART would enhance my effectiveness in learning) asks about learning effectiveness, which may not be appropriate given the mixed audience and intended use of MART as an industry grade tool. As indicated by its relatively high principal component factor of .795, omission of this question may help ameliorate out these two factors versus the Attitude Towards Using MART.

In addition to this single question, the Perceived Usefulness component starts to delve into the actual implementation of MART within the framework of a geographic information infrastructure. While users do have a positive attitude towards MART, questions linger about how it can be implemented on their particular computer system given technical constraints such as current GIS data format, operating system and familiarity or lack thereof with the open source environment. While the open source

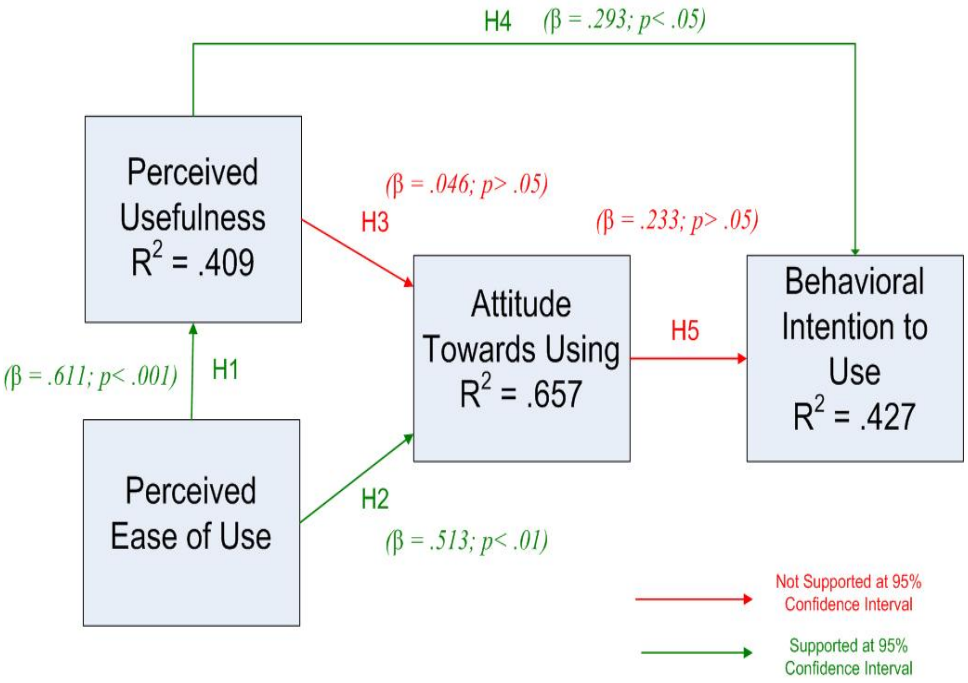

Figure 19. Results from Regression Analysis Used to Test Research Hypotheses

environment does present one slight roadblock when dealing with proprietary GIS data formats (see Discussion), a Windows-based tool can be easily developed for MART using existing techniques performed in this research. Users may not know this and this attitude is reinforced by user comments. As of now, the Attitude Towards Using MART

is determined heavily by the Perceived Ease of Use of MART. This is probably not an indictment of the Perceived Usefulness of MART, but more of a commendation of MART's overwhelming ease of use.

H5 (Attitude towards using MART has a significant effect on intention to use MART) is also not supported at the 95% confidence interval. Possible reasons why this model is not supported is not the intention to use, but the actual implementation of MART given the role of the respondents. As per Table 9, users have a positive attitude towards using MART in the future. However, there seems to be a schism between the assessment presented from a sample database and integrating these assessment tools within the confines of their personal computing environment. Given the responses, another scenario is a planner who utilizes only a few GIS data layers who may not need MART. Given the size of their data sets, their role within the GIS organization and technical expertise, this intention to use dimension may not be able to be modeled within the confines that TAM assessment has presented in this particular research. While precursory analysis relating self-described computer use and GIS experience did not yield any concrete relationships between the Intention to Use MART, the reasoning behind this is probably difficult to encapsulate and bleeds into the fields such as the cognitive analysis into the role of technology in the workplace, our interpretation of this role and how it is managed within the business process. Needless to say, these are topics for further research. Given the t value for this model, it was supported at about a 70% confidence interval. This is promising news.

In summary, TAM analysis proposed a variety of research hypotheses by encapsulating qualitative opinions of MART on a 7-point Likert scale. These questions are grouped based on these research hypotheses and regressed against each other using simple linear models. 3 out the 5 research hypotheses were supported at the 95% confidence interval. Another was supported at about a 70% confidence interval. Similar studies by Marson (2007) where 4 out of the 5 research hypothesis were supported and Hsia et al. (2008) have shown comparable results. Given the revolutionary and burgeoning nature of GIS metadata assessment via MART, these results were extremely promising. While there still needs education in the presentation, implementation and assessment of MART, TAM analysis has affirmed the effectiveness of metadata assessment and the need for information management at higher level than what currently exists.

CHAPTER V

CONCLUSIONS

GIS data in its most primitive form have been in existence since the late 1960s. With the proliferation of desktop and large server computers to efficiently store and process this information, GIS has slowly crept into public consciousness. In today's high-tech world, GIS is implemented in almost all facets of our lives. The UPS driver who delivers our packages uses GIS to optimize his personal route combined with the logistics management required to effectively deliver the package from its origin. The new supermarket being built down the street is there because spatial analysis showed a market based on a variety of different variables could be supported in the neighborhood. With this increased need for GIS, the NSDI was created in 1994 to govern the way in which GIS data are created and managed. In concert with NSDI initiatives, the FGDC has created a framework to help describe the way in which this information was created. It essentially converts the tacit knowledge that we have about a dataset, whether written down somewhere or in our heads to a format that the entire GIS user community can understand and use in the future.

While GIS and the spatial technologies are a rapidly evolving field ripe for research, little research has been performed on the subject of GIS metadata. It is understood that GIS metadata is not as intellectually stimulating as the analysis performed on the GIS data, but is a necessary component for any Geographic Information

System. Metadata serves to validate the data, analysis and maps created as a result of GIS analysis. It is difficult if not impossible to authenticate the results of any GIS analysis without the adequate population of GIS metadata.

While GIS metadata is a subset of the larger field of metadata, inherent differences attributed to the spatial element do exist and as a result, separate software packages are created with the sole purpose of managing GIS metadata. Even less research has been performed on the subject of GIS metadata assessment. Research that has been performed on GIS metadata assessment within the digital environment has limited their applications to a particular popular software package, and therefore a computing environment. Furthermore, metrics to express metadata integrity are few and far between. While this research is valuable to the GIS community, it limits the community's ability to replicate these results if using a different software configuration (software, operating system, patches, updates, etc.). The methodology previously discussed served to fill these technological and intellectual gaps by looking at new applications of information assessment and knowledge discovery within the multitudes of computing environments that exist.

With the information age in full bloom, a higher level of technology is required to assess this information. This stagnant information has no benefits. It is essential that there is a means to find utility in this information and take action from it. However, there are currently few means in which to make this delicate transformation. The goals of the assessment described in this research essentially supported an information infrastructure which inevitably aided in the decision making process. As it pertains to GIS metadata,

the transformation from data into applied knowledge is paramount to any business enterprise and serveed as the impetus of this research.

This research explored methodology to compartmentalize information about this information and delves into the revolutionary field of "meta-metadata". While little research on meta-metadata has been done in the library sciences, even less has been done in the field of GIS. This research made an attempt at GIS meta-metadata through the creation of a XML standard in order to express the data about the metadata. With descriptive statistics about a GIS data set (number of layers), this standard stored high level metrics such as the temporal mean, temporal range, temporal median and average horizontal accuracy about an assessed data set into a new format. This format, named MARTO (Metadata Assessment and Reporting Tool Output), saved information about FGDC metadata compliancy and contained placeholders for links to histograms which conceptualize these metrics and rules derived from unsupervised data mining techniques.

This XML output can be used a variety of different ways by the GIS community. MARTO files run on the same database on a periodic basis (weekly, for example) can be compared to look for changes between the two standard MARTO files. This change detection analysis (CDA) is useful to track changes within a database, both positive and negative, for metadata along some baseline standard. In addition, when allocating resources such as time, personnel for data development efforts, metadata for various datasets can be compared within MARTO files. Users can see which dataset has the poorest temporal accuracy, worst or missing horizontal accuracy and missing metadata elements. Using unsupervised data mining techniques, users can determine if any

deficiencies are related to the data creator, data organization or even location. This analysis gives a cross-database look at specific features within all datasets that need to be revisited. Both a longitudinal and restricted approach to this data analysis serves the GIS community by pinpointing problematic areas in large datasets consisting of perhaps thousands of layers. It would take days or even weeks for a human to replicate these efforts and create such a refined analysis.

Various software packages compete for the rapidly burgeoning market of spatial mapping applications used across many fields. Many of these packages have macro languages that help customize these applications, automate redundant processes or perform new types of analysis within the confines of its software. For Environmental Systems Research Institute (ESRI) software, the Visual Basic programming language using objects specific to the structures and processes used in the creation and analysis of spatial data. For earlier versions of ESRI software, a proprietary object-oriented programming language called Avenue used as this language. In addition, ESRI software also uses a language called Arc Macro Language (AML) as a low level language to replicate commands one would do at a command line and also the Python programming language to perform compound processing routines.

Research by Batcheller (2008) explores how to programmatically access and process these metadata. However, it is done within the confines of ESRI software. This leaves a lot of the GIS community unable to replicate his results for two reasons. GIS data may be stored in a format proprietary to the software and unrecognizable outside of this package. In addition, software languages used by Batheller and others can execute

within only a particular GIS software package and the operating system supported for that software package.

From a practical standpoint, this research focused on the cross-platform programming environments and file formats necessary to perform this meta-metadata analysis regardless of native format. Using XML as the storage medium for GIS metadata, analysis, FGDC compliancy, unsupervised data mining and supervised techniques were performed on any form of GIS metadata. This was accomplished through the open source programming languages of Perl, R and PHP. Some work may be required to convert from the native format on the user side of things. Please refer to the Discussion (see Metadata and Proprietary Formats) for further elaboration.

The Technology Acceptance Model (TAM) has shown that the application of statistical and data mining techniques to GIS metadata has its utility and potential within the GIS user community. These techniques essentially serve as GIS 'meta-metadata', in which users can manage information about large spatial databases so long-term and immediate decisions regarding budgets, personnel and other resources can be made. In this day and age, managing information about GIS data assets can be almost as important as the data themselves. MART served as a viable solution to manage this information. In its most elementary form, these techniques assessed metadata compliance according to FGDC standards. Comments collected with the TAM results have indicated the usefulness of these graphs as adequate tools and metrics to represent GIS metadata integrity not previously seen before. In addition, respondents were able to interact with the data using supervised techniques which can query metadata information from an

entire GIS database. Previously, creating this information would have taken days or even weeks. These techniques cut the pre-processing required to perform these queries down to minutes. In addition, unsupervised techniques glean unseen trends from the GIS data. The supervised techniques were presented in the web where users could query GIS metadata using form elements such as check boxes, text boxes and drop down menus which would access a MySQL database using the PHP programming language. Unsupervised techniques used an existing Perl module were used to create association rules according to a support, or confidence, to see if various metadata elements are related to each other. Only 4 out of the 40 (10%) respondents preferred the unsupervised techniques over the supervised techniques. While the supervised presentation seems more intuitive to a traditional GIS user and provides much greater flexibility, please see the Discussion for further elaboration.

The results from TAM analysis showed that 3 out of the 5 research hypotheses (H1, H2 and H4) relating these assessment and querying technique's Perceived Ease of Use, Perceived Usefulness, Attitude Towards Using and Intention to Use were accepted at a 95% confidence interval. Another (H5) could be accepted at about a 70% confidence interval. In one non-supported hypothesis (H3), it seemed that the strong responses from the Perceived Ease of Use component obfuscated the Perceived Usefulness component. As a result, the Perceived Usefulness did not have much of an effect on the linear regression model used to support the hypothesis. In the other model (H5), the dependent variable is the Intention to Use MART component using the Attitude Towards Using MART as the independent variable. The two components delve into the question of

implementing MART onto their individual system and have elicited a wide array of responses from respondents due to technical experience and familiarity or understanding with the open source environment which are difficult to model. Even modeled by themselves without the Perceived Usefulness component, a linear regression model between these two factors (Attitude Towards Using MART vs. Intention to Use) only produces an R^2 value of .118. Other factors – either those not captured within the assessment or just unquantifiable by their very nature factor into the non-support of this hypothesis. Regardless, these results help satisfy the theoretical impetus of this research and are both intriguing and promising for the future of GIS metadata and widening role as an effective tool to elicit action.

CHAPTER VI

DISCUSSION

MART looked to address the issue of assessing metadata for large GIS databases on two fronts. From a theoretical standpoint, MART attempted to evaluate how different data mining techniques can be applied to GIS metadata. In particular, prior research has shown that the application of unsupervised techniques, in this case association rule learning and different metrics was revolutionary to GIS metadata. Results from respondents have shown its utility. From a practical standpoint, the technologies performed via supervised techniques to query the database have shown to be even more useful to the GIS data community. Using open source programming techniques such as Perl and R to perform processing and PHP to query and display this information via the web, large scale GIS databases were assessed with relative ease. This holistic approach to metadata assessment has shown to save valuable time and resources, while also using data mining techniques to explore unseen trends or patterns within various components of the data.

However, with research of this magnitude, a number of obstacles existed. Open source programming techniques, while useful in a utopian environment, does have its drawbacks in the world of proprietary GIS software and formats. Furthermore, the increasingly improved technologies that aid in GIS data development have outpaced the

development of GIS metadata standards used to catalog these techniques. Some of the roadblocks in applying these ideals in the practical world of GIS are discussed below.

Integrating MART with Remote Sensing Data

MART was catered to standard GIS metadata in the Content Standard for Digital Geospatial Metadata (CSDGM). This standard saves approximately 410 individual features about the GIS data set. The CSGDM is actually derived from ISO (International Standards Organization) Standard 19115. The ISO is an international organization focused on standardizing specifications for almost anything technical across various languages and cultures. A leading role in the proliferation of GIS and the standardization of geospatial technologies was taken on by the United States with the creation of the NSDI (National Spatial Database Infrastructure) in 1994. Given this leading role, many of the requirements dictated by FGDC are actually used in its parent organization, the ISO (Kresse and Fadaie 2004).

Information collected in metadata ranges from the time period in which the data was created to information about the source data in case the data were derived from another data source such as an analog map or digital ortho-imagery created by an independent entity. In addition, detailed information about the person who creates the data, is responsible for the data and completes the metadata documentation is retained. In some cases, different people will assume these roles for a particular data layer. In other cases, it will be the same person. While the FGDC only dictates that seven required and fifteen suggested elements be populated (Table 1) within the confines of metadata, most geospatial enterprises in both industry and government extend these requirements via

proprietary documentation. MART expounds on these requirements to capture some potentially useful information.

As previously mentioned, there exist a number of interdependencies that coincide between metadata elements. Of these 410 individual elements, some include the zip code of the metadata contact or city of the process step contact. These facets of metadata would only be important if the address of the contact were populated. In addition, some of these metadata elements may be thought of as non-essential to a particular geospatial enterprise. Given the use of the GIS dataset, the custom order process or distribution liability may not be applicable and its population is waste of time, and therefore money.

The CSDGM helps to define both the fields, field definition and domain fields for all metadata elements. For the metadata abstract ("a brief narrative summary of the data set"), users are allowed to enter free text. However, for the progress of the data, the CSDGM only allows values of "Complete", "In Work" or "Planned." (FGDC 2000).

While the level of attribution within metadata has improved with each new standard, it is in no way complete. As technologies improve and there include more diverse ways to collect, manipulate and create GIS data, metadata must be flexible enough to accommodate all of these techniques. For example, the standard CSDGM does not contain placeholders germane to the collection of data created via a GPS unit like the various DOP (Dilution of Position) measures such as vertical, horizontal and 3D. In addition, detailed information directly associated with the quality of data such as ephemeris can be entered via a free text field, but lacks the placeholders within the CSDGM. In addition, GIS data extend well beyond the typical raster and data models

that a GIS professional may have solely encountered only a decade earlier. GIS data may now include stand-along tables, TINs (Trianglulated Irregular Networks), relationship classes and even topologies. They each have their own intrinsic qualities that make their creation and update difficult to encapsulate within a single catch-all metadata format.

One way in which these deficiencies are most apparent within the confines of MART is with remotely sensed data. Remote sensing is essentially the collection of data from a distance. It may include terrestrial-based data transmitted via ocean sensors or earthquake monitors, but in the GIS world, remote sensing data generally refers to raster based imagery collected via airborne or space sensors resulting in real or false color images. Remotely sensed data are a subset of raster data. The raster data model is used to model continuous phenomenon that occurs throughout an area. A raster data model would be used to represent elevation, temperature and air pressure. While vector data could represent these facets contours, isotherms or isobars to represent these features using the vector model it is not as accurate (Jensen 1996).

The aforementioned current metadata standard (FGDC Content Standard for Digital Geospatial Metadata) is useful to store metadata for remotely sensed imagery. For any remotely sensed image, it will require a publication date, a primary contact, process step contact, metadata contact, distribution information and even entity attributes if we are using some sort of lookup table for nominal data. This standard even contains a placeholder for cloud cover within the data quality module. However, there is much more processing information that goes into a remotely sensed image that this standard does not store. While useful in nature, remote sensing specialists require more of a

framework so the necessary parameters can be collected about the remotely sensed imagery. Lastly, various image enhancements performed on the imagery extend above and beyond the typical processing steps seen in GIS processing. Some of these steps are performed within the platform itself and others on the ground. There needs to be a way to capture these techniques within these various components.

A need for this standard is no surprise to the remote sensing community. It is a well-known fact that current GIS metadata standards, while useful, merely serve as a subset of expected remote sensing metadata parameters. Dr. Liping Di at the Laboratory for Advanced Information Technology and Standards (LAITS) was one of the first to help formulate standards to capture the necessary parameters to be stored in remotely sensing metadata. While this metadata could have its own ISO standard, this remote sensing metadata standard is an extension of ISO 19115. Its current official name is "Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata" (FGDC 2002). While officially published by the FGDC, Dr. Di served as the project chair for the creation of this metadata standard.

This extension collects information germane to one remotely sensed image, not a set of images. As a result, metadata for imagery taken from the same date can not be applied to all sets of imagery. It is categorized much like current GIS metadata with identification, spatial information and entity information, but also includes places for detailed instrumentation information, ancillary information about the sun, location information and spectral/radiometric properties. At the topmost level, this metadata collects identifying information about the mission, remote sensing platform,

instrumentation name, abstract, date information (temporal accuracy) and access constraints (FGDC 2002).

This extension speaks directly to components that are intrinsic to remotely sensed imagery. There are placeholders for the path and row of the image taken, in addition to the bands and identification of the bands collected for this image. The path and row represent a universal convention to identify location for certain platforms (LandSat, for example) while the bands discuss the spectral resolution of the imagery (Jensen 1996). Users can input the band definitions, stating the range of wavelengths encompassed by each band.

In addition, this extension discusses the processing aspects that go into making this information a remotely sensed image. There are placeholders for the overlap alignment. When a remotely sensed image is taken, it will overlap with a portion of another remotely sensed image. The amount of this overlap varies depending upon location, usually latitude. How much of this overlap occurs and processing steps used to address it (stitch, clip, etc.) are addressed in this metadata extension (FGDC 2002).

Finally, users can input information about the radiometric properties of this image. A field named "Scan_Radiometric_Properties" allows users to input the algorithm to convert quantity in detector units to physical units. Below it is a placeholder for the "Scan_Spectral_Properties" which include the design specifications for scanner properties on a wavelength by wavelength basis. Information about these radiometric and spectral resolutions is stored here, can be edited here and accessed accordingly

(FGDC 2002).

Attribute	Explanation
Mission Name	The character string by which the mission is known.
Mission Start Date	Date that mission during which data were taken began.
Mission Completion	Scheduled or actual end date of mission during which data were taken.
Mission Significant Event	Date and description of a major occurrence during mission.
Full Platform Name	The complete name of the platform. May be different than the mission because multiple platforms could be deployed within the duration of a single mission. For example, LandSat may be a single platform within a mission that may include other platforms.
Platform Information	General information about the platform from which the data were taken.
Platform Start Date	Start date of platform operation.
Number of Bands	Number of separate wavelength ranges at which system measures. This can help delineate between multi-spectral and hyper-spectral.
Ephemeris	Time at which nominal platform orbit or geostationary position is valid. Orbits process and geostationary positions may vary over the life of a platform, and therefore the time of validity must be given. The ephemeris is a table of positions that shows where the platform should be in its orbit.
Band Identification	Complete information to identify instrument wavelengths or other channels. May include the design specifications for properties of an individual wavelength range.
Path	Sequential number, increasing east to west, assigned to satellite orbital track. The path and row coordinates form a unique identifier by which an image can be located via this absolute location
Row	Sequential number assigned to frame latitudinal center line along a path. The path and row coordinates form a unique identifier by which an image can be located via this absolute location
Algorithm Information	Details of the methodology by which geographic information was derived from the instrument readings.
Algorithm Description	Kinds of material providing a description of the algorithm used to generate the data. Used to assist users in understanding what features in their data may arise as a result of the properties of the processing algorithm.
Nadir Latitude	Latitude of nadir. Nadir is defined as point on the surface of the earth directly below the observer.
Nadir Longitude	Longitude of nadir. Nadir is defined as point on the surface of the earth directly below the observer.
Acquisition Information	Error from GMT, needed to correct the time tag for scan data, in milliseconds.
Scan Radiometric Properties	Function used to convert quantity in detector units to physical units. Depending upon the wavelength, a scanner actually detects reflected or emitted energy. This describes the process by which this energy is collected.
Scan Spectral Properties	Design specifications for wavelength-dependent scanner properties. Radiometric energy is converted to spectral metrics via the properties described in this attribute.
Spectral Information	Wavelength-dependent properties of optical systems.
Scan Geometric Properties	Spatial and temporal description of scan.

Table 13. Additional Components that Could be Collected from FGDC Remote Sensing Extension

Lastly, if this remote sensing metadata extension was not enough, there also exists a content standard for remote sensing swath data. In essence, the swath is the length perpendicular to the flight path from which data are collected. While a typical push-broom or whiskbroom sensor uses the swath, other sensors such as the 2D CCD may not. Information collected for this extension includes information about the track and cross-track, as well as the roll, pitch and yaw for the sensor at the time of collection (FGDC 1999). This is very detailed information.

Given the proliferation of remotely sensed data within the field of GIS, this research and ultimately MART would be incomplete without modules to account for these data. Table 13 highlights some of the fields that can be captured from this remote sensing extension and applied to association rule learning or supervised techniques within MART.

Metadata and Proprietary Formats

One of the most obvious and pressing issues associated with this research is the data preparation component. GIS metadata are available in a variety of different formats. In addition to formats compatible with a particular software package such as ESRI, tkme (USGS metadata editor) and CorpsMet (Army Corps of Engineers metadata editor), formats conducive to saving metadata include TXT (Text file), SGML (Standard Generalized Markup Language), HTML (Hyper Text Markup Language) and XML (Extensible Markup Language). Within each of these different formats, metadata information can be organized differently. Standards include FAQ (Frequently Asked

Questions) format, the CSDGM (Content Standard for Digital Geospatial Metadata) or a raw format.

As with the issues discussed with interoperability, it is impossible to predict the type of native data format that an individual GIS user will be using. In addition, designing routines between all of these different formats falls outside of the scope of this research. The programming techniques discussed in this research designed are cross-platform and able to run on any operating system. Whether by GUI (Graphic User Interface) application or an automated program, the same can not be said about all GIS metadata management software. Most, if not all GIS metadata management software, can covert between their native format and XML format. Before running the project associated with this dissertation, users will have to convert their metadata to XML format if they are not working in XML format already.

APPENDIX B shows a sample of VBA/ArcObjects code designed by the researcher and created for ESRI software that mass converts metadata from native format to XML format for all layers in a GIS database. While GIS raster and vector data such as shape files, coverages and GeoTIFFs have GIS metadata already in XML format attached to them, personal geo-database and file geodatabases store metadata in a BLOB (Binary Large Object) field. Unlike numeric (float, double, integer), text or date fields, BLOB fields store complex collections of strings such as metadata using a sequence of binary numbers. Unfortunately, this BLOB format is only understood by ESRI software and a happy medium such as XML is required to large-scale process across all platforms.

While ESRI GIS metadata management software such as ArcCatalog can manually export one layer at a time using its GUI interface, programming constructs such as those highlighted in APPENDIX B automate this process so this export process can be performed for an entire data of dozens or even hundreds of layers at the same time. This code converts all metadata from a selected database to XML format. It concatenates the layer name with the database. The logic behind this was that if two separate county databases contained the same layer (rivers, for example), they could be identified and differentiated in the data mining process and subsequent output. Programmers using other GIS software may need to covert from their proprietary metadata format to this XML standard.

Various Accuracies Within GIS Data

GIS metadata looks to catalog information about all facets of a dataset. In doing so, it helps to encapsulate the life cycle of a GIS data layer and all people and their accompanying contact information that have helped develop the data. In addition, various accuracies that can be quantitatively measured and graphed have been done so in the course of this research.

Within the confines of MART, temporal and horizontal accuracy were captured from metadata and graphed. Relationships between and among these accuracies can be made between other metadata elements or geographically. However, other types of accuracy do exist. Attribute accuracy is a metric to describe to what certainty the descriptive features of a data feature matches its real world representation (Chang 2002). For example, a section of road may be named “Edgar Road” and have a speed limit of 40

miles per hour. Attribute accuracy looks to make sure that the section of road represented within the GIS database does in fact have a speed limit of 40 miles per hour and is named "Edgar Road."

As one can imagine, judging attribute accuracy can be a tedious task. Using the roads feature as an example, a single feature can have many fields to describe it. For a single layer such as a county roads database, it may have thousands of these features. Effectively ensuring that accuracy for each attribute for each feature would be an exercise in futility. By the time one was done verifying the attribute accuracy, the layer would need to be updated once again.

There are techniques to help quantify attribute accuracy. The NRCS (Natural Resources Conservation Service) has devised methodology to help automate the process of "ground-truthing" features within a certain confidence interval, the process of making sure the values in a soils database match that on a map. However, expert knowledge of the attributes, its taxonomy and the study area is required at the human level. Computer applications can help streamline the process, but at the end of the day a human must verify this accuracy (Chang 2002). It is impossible to perform and display this accuracy assessment within the confines of MART without relatively complex human interaction. Future releases of MART may include a module to help assess this attribute accuracy or at least guide this process.

Another form of accuracy relatively new in the field of GIS is logical consistency (FGDC 1998). Logical consistency can encompass many things such as attribute rules and projection compatibility, but for the sake of this research, it describes topology rules

that can be applied to the GIS data. With the new geo-database model used in newer GIS systems not previously implemented with traditional shape files, topology has been an emerging concern for those wishing to perform logical sanity checks on their spatial data. These topological relationships can be amongst a single layer or between multiple GIS data layers. For example, a roads' GIS data layer should not have overlapping features. Two lines on top of each other are difficult to query, visualize and edit. If a concurrency exists, constructs within the attribute information should show that. Violation of this topology rule within the software will flag the error and ask the user what they wish to do. The functionality of these rules is extended for multiple data sets. Using defense GIS data as an example, a firing range for a military installation must be within the installation boundary. While this rule seems logical by nature, this topology error is flagged for a number of reasons. Ranges may be mis-GPSed, their coordinates derived from surveys may not be input correctly or the boundary of the installation may have been re-surveyed so that the range falls outside of the installation boundary.

Regardless, logical consistency as it applies to topology is highly dependent upon the nature of the GIS data in question. As with attribute accuracy, this logical consistency is dependent upon a human element familiar with the GIS data. It is difficult if not impossible to encapsulate this form of accuracy within the confines of MART with a high degree of accuracy at the current time.

One last form of accuracy includes semantic accuracy. Semantic accuracy looks to ensure that the definitions of terms are interchangeable throughout the entire GIS user community (Haunert and Sester 2008). For example, a GIS data layer representing

wetlands may include all swamps, bogs, marshes and fens. However, the national hydrologic dataset created by the USGS may differentiate between these slightly different features and represent each as a separate layer. As a result, swamps, which many may think serve as a majority of wetlands, are truly not representative of all wetlands that another organization may recognize. One agency may define their roads as paved and unpaved surfaces on which vehicular traffic is permitted. Another agency may have separate layers to represent paved and unpaved roads. If merging the roads from these two agencies (counties for example), it is essential to understand the different definitions used by each agency when creating a merged data layer.

With the proliferation of GIS technologies across all cultures and languages, it is increasingly difficult to come to an understanding about what each data layer represents. Semantic accuracy is probably the least explored of all accuracies. MART has no constructs in place to test it. As a result, the abstract, supplemental information and source information should succinctly define the data layer, its meaning and reference to the definition source within the confines of metadata.

The Interestingness Issue of Association Rule Learning

In a sample execution of MART with only 890 GIS data layers, there were more than 6,204 association rules created for support level 2 for a confidence of .7. For the same confidence at support level 4, there were more than 526,000 association rules created! While the multitude of rules as applied to GIS metadata is astounding, one of the issues that association rule learning presents is determining the utility, or interestingness, of these rules.

The job of the mining algorithm is to establish these rules with some established degree of confidence. However, many of these rules may be intuitive in nature or absolutely useless. The computer algorithm does not know that and presents to us, the users, all rules for the appropriate set level and confidence threshold. This is not so much a problem with supervised techniques because users can cut out the minutia themselves. With unsupervised techniques, doing so must be done post-processing.

Helping to pare down the results of these unsupervised techniques has been a focus of research since the first large-scale use of these association rule learning algorithms. Among the first to bring up the issue of quantifying this interestingness among rules was Matheus et al. (1996) who applied association rule learning to inpatient admission data for GTE health systems. They developed KEFIR (Key Findings Report) which summarized changes in large databases and selected various variables on the consultation of health experts on what to do with these changes. Among the findings was that inpatient stays related to mental health were much higher than other stays. Apparently this was not a relationship that was asked of the database; it was later gleaned from data mining techniques. In cases where these and other abnormal situations arose, GTE program managers were consulted on ways to optimally treat these conditions.

Later Lan et al. (2006) tried to improve upon existing techniques to generate effective associative classifiers. Like their predecessors, they assert that intuitive knowledge of the data is the cornerstone for effective association mining, given that adequate support and confidence thresholds be applied to the data. Expounding upon CBA (Classification Based on Associations) algorithms first proposed by Liu et. al in

1998 which uses a metric called 'intensity of implication', they also use a metric called 'dilated chi-square' which tries to find interdependence between condition and class variables. The intensity of implication measures the quality of the association as a function of confidence and support. The dilated chi-squared metric replaces confidence as a means to assess the interestingness of an association. The dilated chi-squared empirically looks at the interdependence between the antecedent (item to the left of the => in a rule) and the consequence (item to the right) to ordinally display the interestingness of a rule. The empirical inspection of rule components decomposed via traditional chi-squared techniques further highlights the research that has gone into methods to trim rules from the output of association rule mining.

Cardinality in the Unsupervised Learning Environment

The unsupervised techniques performed on metadata were an innovative way to approach GIS management. While TAM addressed the issue of its effectiveness within the GIS environment, even less research has been performed on the amount of data that is optimal within the unsupervised data mining environment. Using the testing of 890 data layers, there were 43 different variables collected for each GIS data layer. At support level 4 and a confidence threshold of .7, there were more than 526,000 rules created. Do all of these variables have a positive impact the output of the unsupervised techniques? Is the area encompassed by a data layer really important to elements of metadata? Are the number of features within a data layer or the number of attributes for these features intrinsic to a layer, and therefore unrelated to metadata quality and information gleaned

from metadata? While relationships between these different features may be highly correlated, they may be merely the result of chance.

It can be speculated that the increased cardinality degrades or at least complicates the results of the unsupervised techniques. Cardinality refers to the number of elements in the set, or transaction table in this case. These conceptual hierarchies will exponentially increase the processing time, space and ultimately the number of rules from these unsupervised techniques (Han and Kambler 2005). At what point in time that these attributes begin to become noise is dependent upon a number of factors.

Unsupervised techniques for the 890 data layers were created for only 6 attributes (instead of the previous 43) deemed the most important by the researcher. These attributes are the publication date, data theme, metadata POC, horizontal accuracy, location of data layer (using the latitude and longitude to decide between Northeast, Southeast, Northwest, Southwest, Upper Midwest and Lower Midwest) and the responsible party. With support level 4 and a confidence threshold of .7, there were 2,689 rules resulting from the unsupervised techniques for the 890 datalayers. At support level 2 and a confidence threshold of .7, there were merely 73 rules created. Many of these rules have some use to GIS management. Some of them are listed below:

```
352 0.997 Data_Theme=Public_Safety => Publication_Date=Medium
10 0.714 Location=Upper_Midwest => Publication_Date=New
536 0.747 Horizontal_Accuracy=Unknown => Publication_Date=Medium
```

There are certain tradeoffs when trying to perform unsupervised techniques on GIS data and perhaps this is a case where ‘less is more’ can be a guideline. While using an increased number of attributes will help glean every possible trend from GIS metadata,

the data mining algorithm will produce an increasing number of rules deemed most likely useless by GIS data management. By decreasing this noise, or cardinality by looking at only 6 of the more important metadata elements, the number of rules generated is exponentially decreased.

However, many of the important rules are still retained. It is much easier to delve through this set of rules based on these important metadata attributes rather than creating an all-inclusive data set that still requires the human element. These results can be combined with the queries of supervised techniques to efficiently extract information from the multitude of metadata information. Regardless, GIS managers are the ultimate decision maker in this process and must find a happy medium between creating adequate results, superfluous output and the resources dedicated towards discriminating between the two.

The TAM Methodology

Many TAM analyses have been performed to test the acceptance of technology within a larger user community. Theses analyses contain a wide range of models and even a wider range of questions and groups of questions (components). Some earlier models are strictly linear, meaning that a linear regression between only one component is used to adequately explain another dependent single component. These types of models are easy to see and visualize. The questions and model used in this TAM analysis are based on a study by Masron (2007) due to its content and relationship to the e-learning environment. In two cases, multiple linear regression is performed on two independent variables to explain a dependent variable. The contribution of these

independent variables to this model support or contradict the research hypothesis. Given the dependence of these components and the questions that compose them, researchers must be mindful about the questionnaire used to solicit opinions. It was previously surmised that question USE1 may not be fitting given the intended audience. That, combined with users' high opinion of the Perceived Ease of Use component may cause the model to be entirely dependent upon one variable. That seems to be a case with H2 and H3. Running a Perceived Usefulness Model vs. Attitude Towards Using alone yields favorable results ($\beta = .360$, $p < .01$). Setting up a slightly different research model or re-wording the questions may increase the contribution that Perceived Usefulness has towards the Attitude Towards Using in this multivariate linear regression model.

Presentation of Unsupervised Techniques

Part of the intellectual impetus of this research was the application of large scale analysis and data mining techniques to GIS metadata. This was done using both supervised and unsupervised techniques. Using supervised techniques, a web form was created so users could select fields of metadata they wished to query. They could find all records that had the word 'water' in the abstract, query all publication dates that were unknown, or even both of these at the same time. In doing so, a flexible and dynamic web presentation tool was created to users could ask a variety of questions about the GIS metadata.

However, the results from unsupervised techniques are presented in a static text file with the relationship and its support. Depending upon the dataset used to create the rules, this text file may be extremely large in size and the data presented in it has no

apparent order. While this information is interesting, the plethora of data may be overwhelming to lay users. Much like the supervised techniques, a web presentation tool to present the results of unsupervised learning may be useful for a project of this magnitude and could be the subject for further research.

The Open Source Environment

Finally, in the TAM analysis, the Perceived Usefulness and Intention to Use Component delved into the actual implementation of MART onto a host system. This includes Perl, R, PHP and even the MySQL database used to compose and later query the data. MART can run once it has been installed on a single host computer which has access to the GIS data intended for assessment. While open source has many connotations, for all intents and purposes, open source refers to 1) software that can run under any operating system and 2) is free of charge. The software used in all facets of this research satisfies both of these stipulations. Given the researcher's extensive experience with ESRI software, it would have been easiest to create a custom application to perform this analysis within the Windows environment. However, this would exclude a lot of the GIS user community who don't use ESRI software or lack the necessary technical requirements (version or proper software updates) to run this proprietary software.

Besides the open source software, no extra plug-ins or installations are required. Within the computing community and reinforced by the user comments, there may be a misunderstanding or lack of understanding about the open source environment. Open source applications do not run parallel to a traditional Windows environment, but are now integrated into popular operating systems. While open source is not proprietary in nature

and can not run with specific GIS applications, open source applications can open and read most types of files used to store metadata, such as TXT and in the case of MART, XML. Hopefully users can get a better understanding and break down any misconceptions of the open source environment.

REFERENCES

American National Standards Institute. 1975. *Representations of Universal Time, Local Time Differentials, and United States Time Zone Reference for Information Interchange* (ANSI X3.51-1975): New York, American National Standards Institute.

Anderson, J.D., Perez-Carballo, J., 2001. The Nature of Indexing: How Humans and Machines Analyze Messages and Texts for Retrieval: Part I: Research, and the Nature of Human Indexing. *Information Processing and Management: An International Journal* 37 (2): 231–254.

Aref, W.G. and H. Samet. 1991. Optimization Strategies for Spatial Query Processing. *Proceedings of the 17th International Conference of VLDB*. Barcelona, Spail.

Bandura, A. 1982. Self-Efficacy Mechanism in Human Agency. *American Psychologist* 37(2): 122-147.

Bagozzi, R. P., Davis, F. D., & Warshaw, P. R. 1992. Development and Test of a Theory of Technological Learning and Usage. *Human Relations*, 45(7): 660-686.

Bao, Shuming, Henry, M.S, Barkley, D. and K. Brooks. 1995. RAS: A Regional Analysis System Integrated with Arc/INFO. *Computing, Environment and Urban Systems* 19 (1): 37 – 56.

Bell, D.A., S.S. Anand and C.M. Shapcott. 1994. Database Mining in Spatial Databases. *International Workshop on Spatial-Temporal Databases*.

Batcheller, James. 2008. Automating Geospatial Metadata Generation—An Integrated Data Management and Documentation Approach. *Computers & Geosciences* 34: 387 – 398.

Bruce T. and D. Hillman. 2004. The Continuum of Metadata Quality: Defining, Expressing, Exploiting. In: D. Hillman and E. Westbrooks, Editors, *Metadata in Practice*, ALA Editions, Chicago, p. 238–256.

Chang, Kang-Tsung. 2002. *Introduction to Geographic Information Systems*. New York, NY: McGraw Hill Higher Education.

Coleman, D. J., & McLaughlin, J. D. 1998. Defining Global Geospatial Data Infrastructure (GGDI): Components, Stakeholders and Interfaces. *Geomatica* 52: 129–144.

Craven, T. 2001. DESCRIPTION Meta Tags in Public Home and Linked pages. *LIBRES: Library and Information Science Research Electronic Journal* 11 (2).

Dasgupta, S., Granger, M. & Mcgarry, N. 2002. User Acceptance of E-collaboration Technology: An Extension of the Technology Acceptance Model. *Group Decision and Negotiation* 11: 87-100.

Davis, F. D. 1989. Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. *MIS Quarterly* 13(3): 319-340.

Dimitrova, D. V. & Chen, Y. C. (2006). Profiling the Adopters of E-government Information and Services: The Influence of Psychological Characteristics, Civic Mindedness, and Information Channels. *Social Science Computer Review* 24(2): 172-188.

Doctorow, C. 2001. Metacrap: Putting the Torch to Seven Straw-Men of the Meta-Utopia. http://www.well.com/~doctorow/metacrap.htm#0 (last accessed 29 February 2008).

Drake, Miriam. 2004. *Encyclopedia of Library and Information Science, 2nd Edition.* New York: Marcel Dekker.

Dramowicz, Ela and C. Dramowicz. 2008. Choropleth Mapping with Exploratory Data Analysis. Directions Magazine [website]. http://www.directionsmag.com/article.php?article_id=718&trv=1 (last accessed 20 August 2008).

Dublin Core. 2008. *Dublin Core Metadata Element Set, Version 1.1: Reference Description*. Dublin Core Metadata Initiative [website]. http://www.dublincore.org (last accessed 8 February 2008)

ESRI ArcGIS 9.0 Help Documentation. 2004a. *Geodatabase Items*. ArcGIS Desktop Help (last accessed 27 December 2007).

ESRI ArcGIS 9.0 Help Documentation. 2004b. *Supported Raster Formats*. ArcGIS Desktop Help (last accessed 27 December 2007).

ESRI. 2005. *ESRI/SAP News From Around the Globe*. Environmental Systems Research Corporate Alliance Program [website]. http://www.esri.com/partners /alliances/sap/sapnews/global_news_1104.html (last accessed 7 April 2006).

Fayad, U. G. Pietetsky-Shapiro, P. Smyth and R. Uthurusamy. 1996. *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: MIT Press.

FGDC (Federal Geographic Data Committee). 1996. *National Standard for Spatial Data Accuracy*. Washington D.C.: Federal Geographic Data Committee.

FGDC (Federal Geographic Data Committee). 1998. *Content Standard for Digital Geospatial Metadata*. Washington D.C.: Federal Geographic Data Committee.

FGDC (Federal Geographic Data Committee). 1999. *The FGDC Content Standard for Digital Geospatial Metadata: Content Standard for Remote Sensing Swath Data (FGDC-STD-009-1999)*. Washington D.C.: Federal Geographic Data Committee.

FGDC (Federal Geographic Data Committee). 2000. *Content Standard for Digital Geospatial Metadata Workbook*. Washington D.C.: Federal Geographic Data Committee.

FGDC (Federal Geographic Data Committee). 2002. *Content Standard for Digital Metadata: Extensions for Remote Sensing Data*. Washington D.C.: Federal Geographic Data Committee.

Fishbein, M. & Ajzen, I. 1975. *Belief, Attitude Intention and Behavior: An Introduction to Theory and Research*. Reading, MA: Addison-Wesley.

Frankowski, Dan. 2008. *Data::Mining:AssociationRules – Mine Association Rules and Frequent Data Sets.* CPAN (Comprehensive Perl Archive Network) [website]. http://cpansearch.perl.org/src/DFRANKOW/Data-Mining-AssociationRules-0.10/README.htm (last accessed 23 February 2009)

Gefen, David, Elena Karahanna and Detmar W. Straub. 2003. Trust and TAM in Online Shopping: An Integrated Model. *MIS Quarterly* 27(1): 51 – 92.

Geng, Liquang and H. Hamilton. 2006. Interestingness Measures for Data Mining: A Survey. *ACM Computing Surveys* 38(3).

Gibson, James. 2008. *Geo::Ellipsoid – Latitude and Longitude Calculations Using the Ellipsoid Model*. CPAN (Comprehensive Perl Archive Network) [website]. http://search.cpan.org/~jgibson/Geo-Ellipsoid-1.12/lib/Geo/Ellipsoid.pm (last accessed 27 February 2009)

Green, David and T. Bossomaier. 2002. *Online GIS and Spatial Metadata*. New York: Taylor and Francis.

Greenberg, J., Spurgin, K., Crystal, A.. 2006. Functionalities for Automatic Metadata Generation Applications: A Survey of Metadata Experts' Opinions. *International Journal of Metadata, Semantics and Ontologies* 1 (1): 3–20.

Grubisec, Tony and M. Zook. 2007. A Ticket to Ride: Evolving Landscapes of Air Travel Accessibility in the United States. *Journal of Transport Geography* 15: 417 – 430.

Hair, J.F. Jr. , Anderson, R.E., Tatham, R.L., & Black, W.C. 1998. *Multivariate Data Analysis, (5th Edition)*. Upper Saddle River, NJ: Prentice Hall.

Han, Jiawei and Micheline Kamber. 2005. *Data Mining: Concepts and Techniques*. New York: Morgan Kaufmann.

Hass, Stephanie, E. Henjum, M. O'Daniel and J. Aufmuth. 2003. Darwin and MARC: A Voyage of Metadata Discovery. *Library Collections, Acquisitions, & Technical Services* 27: 291 – 304.

Haunert, J.-H. and M. Sester. 2008. Assuring Logical Consistency and Semantic Accuracy in Map Generalization. *Photogrammetrie - Fernerkundung - Geoinformation (PFG)* 3: 65-173.

Hoaglin, D.C, Mosteller, F and J. Tukey. 1983. *Understanding Robust and Data Exploratory Analysis*. New York: Springer.

Hsia Jung-Wen, Ai-Hua Tseng, 2008. *An Enhanced Technology Acceptance Model for E-Learning Systems in High-Tech Companies in Taiwan: Analyzed by Structural Equation Modeling*, cw, pp.39-44, 2008 International Conference on Cyberworlds.

Hufnagel, E. M. and Conca, C. 1994. Use Response Data: The Potential for Errors and Biases. *Information Systems Research* 5: 48-73.

Jenkins, M. & Hanson, J. 2003. *E-Learning Series: A Guide for Senior Managers*. Coventry, England: Learning and Teaching Support Network (LSTN) Generic Centre.

Jensen, John. 1996. *Introductory Digital Image Processing: A Remote Sensing Perspective*. Saddle River, NJ: Prentice Hall.

Kazar, B. M., , S. Shekhar, D. J. Lilja, R. R. Vatsavai, R. K. Pace, Comparing Exact and Approximate Spatial Auto-Regression Model Solutions for Spatial Data Analysis, *Proc. of Third International Conference on Geographic Information Science (GIScience2004)*, Maryland, USA, October 2004.

Klösgen, Willi and J. Żytkow. 2002. *Handbook of Data Mining and Knowledge Discovery*. Oxford: Oxford University Press.

Koperski, K. and J. Han,. 1996a. Data Mining Methods for the Analysis of Large Geographic Databases. *Proc. 10th Annual Conference on GIS*, Vancouver, Canada, March 1996.

Koperski, K., J. Adhikary and J. Han. 1996b. Spatial Data Mining: Progress and Challenges, *1996 SIGMOD'96 Workshop. on Research Issues on Data Mining and Knowledge Discovery (DMKD'96)*, Montreal, Canada.

Koufaris, M. 2002. Applying the Technology Acceptance Model and Flow Theory to Online Consumer Behavior. *Information Systems Research* 13 (2): 205-223.

Kresse, Wolfgang and K. Fadaie. 2004. *ISO Standards for Geographic Information*. New York: Springer.

Lan, Yu, D. Janssens, G. Chen and Geert Wets. 2006. Improving Associative Classification by Incorporating Novel Interestingness Measures. *Expert Systems with Applications* 31(1): 184 - 192.

Lanter, D.P. 1993. A Lineage Meta-Database Approach Towards Spatial Analytic Database Optimization. *Cartography and Geographic Information Systems* 20(2): 112-121.

Lanter, D.P. 1994.. The Contribution of ARC/INFO's Log File to Metadata Analysis of GIS Data Processing, Proceedings of the Fourteenth Annual ESRI User Conference, Palm Springs, California.

Lee, Ming-Che, K. Hua Tsai and Tzone I. Wang. 2008. A Practical Ontology Query Expansion Algorithm For Semantic-Aware Learning Objects Retrieval. *Computers & Education* 50(4): 1240-1257.

Leiden, K., Laughery, K.R., Keller, J., French, J., Warwick, W., Wood, S.D.. 2001. *A Review of Human Performance Models for the Prediction of Human Error.* Moffett Field, CA : National Aeronautics and Space Administration.

Library of Congress. 2008. *MARC Standards*. Library of Congress Network Development and MARC Standards [website]. http://www.loc.gov/marc/ (last accessed 26 February 2008).

Limbach, T., Krawczyk, A., Surowiec, G. 2004. Metadata Lifecycle Management with GIS context. Proceedings of the 10th EC GI & GIS Workshop, ESDI State of the Art. Warsaw, Poland.

Liu, B., Hsu, W., & Ma, Y. 1998. *Integrating Classification and Association Rule Mining*. Proceedings of the Fourth International Conference on Discovery and Data Mining, New York, US: 80–86.

Malhotra, Yogesh Dennis F. Galletta. 1999. *Extending the Technology Acceptance Model to Account for Social Influence: Theoretical Bases and Empirical Validation*. Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 1, 1999: 1006.

Masrom, Maslin. 2007. *Technology Acceptance Model and E-learning.* In: 12th International Conference on Education, May 21 - 24, Sultan Hassanal Bolkiah Institute of Education, Universiti Brunei Darussalam.

Matheus, Christopher J., Gregory Piatetsky-Shapiro, Dwight McNeill. 1996. Selecting and Reporting What Is Interesting. *Advances in Knowledge Discovery and Data Mining*: 495-515.

McCabe, Thomas. 1976. A Complexity Measure. *IEEE Transactions on Software Engineering* 2(4): 308 – 320.

McLean, Thomas R., L. Burton, C. Haller, P. McLean. 2008. Electronic Medical Record Metadata: Uses and Liability. *Journal of the American College of Surgeons* 206(3): 405 – 411.

Meyers, Lawrence S.; Anthony Guarino, Glenn Gamst. 2005. *Applied Multivariate Research: Design and Interpretation*. Thousand Oaks, CA: Sage Publications.

Monmonier, Mark. 1997. *Cartographies of Danger*. Chicago, IL: University of Chicago Press.

Neimi, Nathan. 2002. *Join Item After SDTS Import.* ESRI Support Home Web Page [website]. http://forums.esri.com/Thread.asp?c=93&f=1149&t=80948#218165 (last accessed 1 March 2008).

Nedovic-Budic, Zorica, M.E. Feeney, A. Rajabifard, I. Williamson. 2004. Are SDIs serving the needs of local planning? Case study of Victoria, Australia and Illinois, USA. *Computers, Environment and Urban Systems* 28(4): 320 – 351.

New York State Office of Cyber Security and Critical Infrastructure Coordination. 2006. New York State GIS Clearinghouse [website]. http://state.ny.gov/gis (last accessed 18 December 2007).

Nunnally, J.C. 1978. *Psychometric Theory*, (2nd ed.). New York: McGraw-Hill.

Ott, R.L., 1993. *An Introduction to Statistical Methods and Data Analysis*. Belmont: Duxbury Press.

Qi, L., Lingling, G., Feng, H., Yong, T. 2004. A Unified Metadata Information Management Framework For the Digital City. *Proceedings of IEEE's Geoscience and Remote Sensing Symposium*, Anchorage, Alaska, USA: 4422–4424.

Reese, T. 2005. Bibliographic Freedom and the Future Direction of Map Cataloging. *Journal of Map and Geography Libraries* 2(1): 67-90.

Reichardt, Mark. 2005. The Dangers of Non-Interoperability. *Earth Imaging Journal* 2(1): 22 - 24.

Robey, D. 1979. User Attitudes and Management In- formation System Use. *Academy of Management Journal* 22(3): 527- 538.

Rogers, E.M. and Shoemaker, F.F. 1971. *Communication of Innovations: A Cross-Cultural Approach*. New York, NY: Free Press. .

Schultz, R.L. and Slevin, D.P. 1975. Implementation and Organizational Validity: An Empirical Investigation. *Implementing Operations Research/Management Science*. , New York: American Elsevier, NY: 153-182.

Schwartz, C. 2002. *Sorting Out the Web: Approaches to Subject Access*. Westport, Connecticut: Ablex Publishing.

Schwartz, Randall, T. Christiansen, B.D. Foy and L. Wall. 2005. Learning Perl. Cambridge, MA: O'Reilly Media.

Sevcik, Peter. March 2002. The Pitfalls of Scaling VOIP. *Business Communications Review*. [website] http://www.netforecast.com/Articles/BCR%20C20%20Pitfalls%20of%20Scaling%20VoIP%20FNL.pdf

Skågeby, Jörgen. 2008. Semi-Public End-User Content Contributions—A Case-Study of Concerns and Intentions in Online Photo-Sharing. *International Journal of Human-Computer Studies* 66(4): *287-300.*

Spielman, Seth and J.C. Thill. 2008. Social Area Analysis, Data Mining, and GIS. *Computers, Environment and Urban Systems* 32: 110 – 122.

Steinberg, Steven and S. Steinberg. 2006. Geographic Information Systems for the Social Sciences: Investigating Place and Space. Thousand Oaks, CA: Sage Publishing.

Stewart, T. 1986. Task Fit, Ease-of-Use and Computer facilities. In N. Bjørn-Andersen, K. Eason, & D. Robey (Eds.), *Managing Computer Impact: An international Study of Management and Organizations* (pp. 63-76). Norwood, NJ: Ablex.

Stvilia, Besiki and L. Gasser. 2008. Value-Based Metadata Quality Assessment. *Library & Information Science Research* 30(1): p. 67 – 74.

Taipale, K. A. 2007. The Privacy Implications of Government Data Mining Programs, Testimony before the U.S. Senate Committee on the Judiciary, Washington, DC (10 January 2007).

Theodosiou, Theodosios, L. Angelis and Athena Vakali. 2008. Non-Linear Correlation of Content and Metadata Information Extracted From Biomedical Article Datasets. Journal of Biomedical Informatics 41(1): 202 – 216.

Tobler, W. R. 1970. A Computer Model Simulation of Urban Growth in the Detroit Region. Economic Geography 46(2): 234-240.

Tornatzky, L.G. and Klein, K.J. 1982. Innovation Char- acteristics and Innovation Adoption-Implementation: A Meta-Analysis of Findings. *IEEE Transactions on Engineering Management* 29(1): 28-45.

Tsou, Ming-Hsiang. 2002. *An Operational Metadata Framework for Searching, Indexing and Retrieving Distributed Geographic Information Services on the Internet*. [website]. http://typhoon.sdsu.edu/tsou/index.html (last accessed 12 December 2007).

Tuchyna, Martin. 2006. Establishment of Spatial Data Infrastructure Within the Environmental Sector in Slovak Republic. *Environmental Modeling & Software* 21: 572 – 1578.

Ullman, Larry. 2008. *PHP 6 and MySQL 5*. Berkeley, CA: Peachpit Press.

USGS. 2005. *Overview of SDST Document*. Spatial Data Transfer System [website]. http://mcmcweb.er.usgs.gov/sdts/standard.html (last accessed 16 December 2007).

USGS. 1947. *National Mapping Accuracy Standards*. National Geospatial Program Standards [website]. http://rockyweb.cr.usgs.gov/nmpstds/acrodocs/nmas/NMAS647.PDF (last accessed 28 February 2008).

Venables W.N et al. 2006. *An Introduction to R (Version 2.3.0)*. Bristol, United Kingdom: R Development Core Team – Network Theory Limited Publishing.

West, Lawrence and T. Hess. 2001. Metadata as a Knowledge Management Tool: Supporting Intelligent Agent and End User Access to Spatial Data. *Decision Support Systems* 32: 247 – 264.

White, N.E.T. 2002. The Mean Opinion Score: Rating Standards for Voice Communications. Voice Quality Evaluation for Communications Systems. [website] http://www.net.com/products/narrowband/repository/white_papers/mos_wp/home.shtml

Wong, D.W.S., Wu, C.V. 1996. *Spatial Metadata and GIS for Decision Support***.** Proceedings of the Twenty-Ninth Hawaii International Conference. Volume 3 (3 – 6): 557 – 566.

Yao, Xiaobai. 2007. Where are Public Transit Needed – Examining Potential Demand for Public Transit for Commuting Trips. *Computers, Environment and Urban Systems* 31: 535 – 550.

Zimmerman, Dale and C. Pavlik. 2008. Quantifying the Effects of Mask Metadata Disclosure and Multiple Releases on the Confidentiality of Geographically Masked Health Data. *Geographical Analysis* 40(1): 52 – 76.

APPENDIX A: INDIVIDUAL RESPONSES FROM QUESTIONAIRE

Table 14. Output from TAM Analysis

	Perceived Ease of Use (PEOU)				Perceived Usefulness (PU)				Attitude Towards Using (ATTITUDE)				Intention to Use (ITU)									
Date	EASE1	EASE2	EASE3	EASE4	USE1	USE2	USE3	USE4	ATT1	ATT2	ATT3	ATT4	INT1	INT2	INT3	AGE	SEX	TITLE	EXP	COMPUTER USE	GIS USE	DATA DEVELOPMENT	META-DATA
6/10/2009	6	6	7	6	6	5	6	6	5	5	5	5	5	3	5	31	F	GIS Analyst	4	32.00	30.00	10.00	1.00
6/9/2009	6	5	5	4	5	5	4	4	5	5	3	5	3	3	3	47	F	Asset Manager	4	40.00	35.00	10.00	1.00
6/9/2009	6	5	6	6	5	4	4	5	6	4	5	4	4	3	4	47	M	Adjunct Professor	4	35.00	20.00	5.00	1.00
7/2/2009	6	6	7	6	7	6	6	6	5	5	4	6	5	4	5	37	M	Assistant Professor	5	36.00	25.00	5.00	0.00
6/23/2009	6	5	5	6	5	5	5	6	5	5	4	5	5	4	4	25	F	Associate support engineer	4	40.00	25.00	5.00	1.00
6/10/2009	6	6	6	6	5	5	5	5	5	6	5	5	3	3	3	29	M	Crime Analyst	4	30.00	12.00	4.00	1.00
6/25/2009	6	7	6	5	7	7	7	7	6	5	5	5	5	4	4	28	F	Cultural Resources GIS Coordiator	4	30.00	24.00	10.00	0.00
6/9/2009	5	5	5	5	7	7	6	6	5	5	7	7	5	2	2	36	M	Engineer	4	40.00	25.00	15.00	0.00
6/12/2009	5	6	5	6	5	5	5	5	5	5	4	5	5	4	4	26	M	G.I.S. Technician	4	30.00	25.00	16.00	1.00
6/16/2009	5	6	6	7	6	6	6	6	4	4	4	4	4	3	4	43	M	GIS & Public Safety Administrator	4	35.00	15.00	0.00	0.00
6/10/2009	7	7	7	6	7	7	7	7	5	5	7	7	5	3	4	34	F	GIS Administrator	5	40.00	30.00	20.00	2.00
6/1/2009	7	6	6	6	7	6	6	7	7	5	5	6	5	6	5	35	M	GIS Analyst	5	35.00	20.00	5.00	1.00
6/22/2009	6	6	6	6	5	5	6	5	7	5	5	6	4	3	4	34	M	GIS Analyst	4	35.00	25.00	8.00	0.00
6/29/2009	5	4	5	4	5	5	5	5	4	4	4	4	4	4	4		F	GIS Analyst	4				
6/16/2009	6	6	5	6	6	5	6	6	5	5	5	5	4	4	4	53	F	GIS Coordinator	5	36.00	25.00	5.00	0.00
6/19/2009	6	5	6	6	5	5	5	5	6	7	7	6	4	4	4	29	F	GIS Director and Research Analyst	4	30.00	25.00	12.00	0.00
6/9/2009	5	4	5	5	5	4	5	5	5	5	5	5	4	4	5	31	F	GIS Instructor	4	35.00	20.00	10.00	0.00

6/17/2009	6	6	6	6	7	7	6	7	5	5	4	5	4	4	4	23	F	GIS Intern	3	35.00	30.00	10.00	1.00
6/9/2009	4	4	7	6	7	7	7	7	4	5	5	5	5	3	2	23	F	GIS Planner	3	22.00	15.00	10.00	0.00
6/15/2009	5	4	6	5	5	5	5	5	5	4	4	5	5	4	4	27	M	GIS Tech	4	30.00	20.00	10.00	0.00
6/12/2009	6	6	7	6	5	5	5	5	5	5	5	5	4	3	4	29	M	GIS Technician	3	35.00	30.00	10.00	1.00
6/29/2009	6	6	5	2	5	5	5	5	5	4	4	5	3	3	3	27	M	GIS/Planner Technician	3	30.00	20.00	10.00	1.00
6/9/2009	7	7	6	7	6	4	5	6	5	6	6	7	5	4	2	24	M	grad student	5	20.00	20.00	0.00	0.00
7/6/2009	6	7	7	6	6	0	6	6	5	5	5	5	6	5	6	24	M	Graduate Student	4	30.00	18.00	4.00	0.00
6/10/2009	6	6	7	5	5	5	5	5	5	5	7	7	4	4	4	44	M	Housing Director	3	32.00	4.00	0.00	0.00
7/6/2009	6	6	6	5	6	6	6	5	5	5	5	6	5	3	4	54	M	Lab Manager	4	36.00	20.00	8.00	1.00
6/29/2009	5	5	5	5	5	5	5	5	5	4	4	4	3	3	3	40	M	Planner	3	30.00	12.00	0.00	0.00
6/9/2009	7	7	5	3	7	6	5	5	7	5	4	7	5	4	5	43	F	Planner	4	30.00	16.00	5.00	0.50
7/1/2009	5	5	4	5	5	4	5	4	4	4	4	4	3	3	3	32	M	Planner II	3	35.00	8.00	0.00	0.00
6/23/2009	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	47	M	Planning Director	3	30.00	5.00	0.00	0.00
7/1/2009	7	7	7	6	5	7	7	6	7	5	5	5	6	4	4	44	F	Planning Technician	4	35.00	30.00	25.00	0.00
6/10/2009	7	7	7	6	6	7	7	6	7	5	6	7	5	6	6	43	M	Project Manager	4	35.00	20.00	12.00	0.00
6/9/2009	5	5	5	5	4	5	4	5	5	5	5	5	2	2	2	41	F	Project Manager	4	36.00	10.00	0.00	0.00
6/10/2009	6	5	6	5	5	5	4	4	5	5	6	5	5	4	4	41	M	Research & Evaluation Analyst	3	40.00	5.00	1.00	0.00
7/1/2009	5	5	6	5	5	5	5	5	4	4	4	3	4	4	4	41	F	Senior Consultant	3	30.00	10.00	0.00	0.00
7/6/2009	3	3	2	3	3	3	3	3	4	4	4	4	2	2	2	49	F	Sr. Engineering Technician	4	35.00	20.00	5.00	0.00
6/9/2009	7	7	6	7	7	7	7	6	6	6	6	5	5	4	4	38	M	Statistician II	3	35.00	10.00	0.00	0.00
6/1/2009	7	6	6	6	7	5	5	5	5	6	6	6	4	4	4	31	F	Student	3	20.00	10.00	1.00	0.00
6/10/2009	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	51	F	Supervisor of Land Records	4	25.00	5.00	0.00	0.00
6/3/2009	7	6	6	5	6	6	6	7	6	4	4	5	4	3	4	24	F	Technician	2	20.00	15.00	2.00	0.00

APPENDIX B: EXAMPLE OF VBA/ARCOBJECTS CODE

This VBA/ArcObjects code is used to mass convert metadata saved in the personal or file geodatabase format, proprietary to ESRI software, to XML files.

```
Private Sub XML_Click()

Update of mass XML export.  Be sure to update in Appendix A

' This application toggle through layers and will export the
' metadata from proprietary format to XML format
Dim pGxApp As IGxApplication
Set pGxApp = Application
Dim pGxObject As IGxObject
Set pGxObject = pGxApp.SelectedObject
' Checks to see if this is a Geodatabase.  Otherwise, the application will close
If Not TypeOf pGxObject Is IGxDatabase Then
    MsgBox "GeoDatabase Not Selected!"
    Exit Sub
End If
' Varible declaration
Dim pGxDatabase As IGxDatabase
Dim pWorkspace As IWorkspace
Dim pDatasetNames As IEnumDatasetName
Dim pDatasetName As IDatasetName
Dim pMetadata As IMetadata
Dim pPropertySet As IPropertySet
Dim pName As IName
Dim pSubSets As IEnumDatasetName
Dim pSubset As IDatasetName
Dim outputfilename As String
Dim outputFGDCname As String
Dim pExportXML As ExportXML
Dim pExport As IMetadataExport
```

```
Set pGxDatabase = pGxObject
Set pWorkspace = pGxDatabase.Workspace
Set pDatasetNames = pWorkspace.DatasetNames(esriDTAny)
Set pDatasetName = pDatasetNames.Next
Do While Not pDatasetName Is Nothing
Debug.Print pDatasetName.Name
'Catch standalone featureclasses
If pDatasetName.Type = esriDTFeatureClass Then
    Set pName = pDatasetName
    Set pMetadata = pName.Open
    Set pPropertySet = pMetadata.Metadata

'****Do Something if layer is in stand alone feature class with no feature metadata set.....
'****Our data should not have any of these stand along feauture classes

    ElseIf pDatasetName.Type = esriDTFeatureDataset Then
    Set pSubSets = pDatasetName.SubsetNames
    Set pSubset = pSubSets.Next
    Do While Not pSubset Is Nothing

    If pSubset.Type = esriDTFeatureClass Then

  '      Debug.Print pSubset.Name
        Set pName = pSubset
        Set pMetadata = pName.Open

        ' Loop stuff in here

        Set pExportXML = New ExportXML
        Set pExport = pExportXML

 '       MsgBox pSubset.Name & pGxObject.Name

        ' This will concatonate the name of the layer with the database name
        ' so a unique layer name will appear and mined within the hundreds or
        ' thousands of layers in this project
        pExport.Export pMetadata, "C:\Metadata\" & pSubset.Name & "_" & Left(pGxObject.Name,
```

```
Len(pGxObject.Name) - 4) & ".xml"

    End If
        Set pSubset = pSubSets.Next
    Loop
End If
' Go to the next feature data set
Set pDatasetName = pDatasetNames.Next
Loop

End Sub
```

APPENDIX C: PERL CODE USED TO CONSOLIDATE XML DATA

This file performs the following functions:

1. Opens separate XML files from location and writes values to CSV file
2. Open GIS related Perl functions to compute geodetically correct areas and distances to be populated.
3. Checks FGDC compliancy
4. Creates HTML file to show compliance to FGDC required and suggested features .
5. Use in Association Rule Mining.

```
# Extract CSV
# Timothy Mulrooney      Dissertation Research        University of North Carolina, Greensboro
#
# This program is designed to prepare data for extraction and
# analysis.  XML data from many files will be converted to a single file.
# The purpose of this program is to extract elements from XML medata
# and place these elements into a CSV (Comma Separated Values) file
# format.  From this format, large scale data anlaysis and data mining
# can be performed on the data.
#
# The command at the prompt should be:
# perl extract_csv_tjm.pl -s c:\perl_test -o test.csv
#  -s --> source (where the xml files are located)
#  -o --> output file (csv file that data will be written to
#  -r --> Name of the report file
#  -t --> Name of the transaction file
use strict;
use XML::Simple;

use Getopt::Long;
use Geo::Ellipsoid;
use Date::Manip qw(ParseDate UnixDate);
use File::Basename;
use File::Copy;
use File::Temp  qw(tempfile);

use warnings;
```

```
use diagnostics;

# ######### Variable Declaration
my $meters_in_a_mile = 1609.347219;
my $miles_in_a_kilometer = 0.6213699;
#my $sfile;
#my $mvalue;

# Defaults if no parameters are entered at the command line
my $sourceDir = ".";
my $output = './test.csv';
my $report = './report.html';
my $transaction = './transaction.txt';

my $filename;
my $transaction_number = 0;
my $version = "PEMT 1.0, 2/24/2009 \n";
my %metadata;   # hash object used to store metadata information extracted from XML
my $key;
my $foundfile = 0;
my @files;

# Array that contains the names of the missing elements required FGDC required and suggested elements
my @rMissing = ();
my @sMissing = ();
my $options;
my $tree;
my $Item;
my $ns_distance_mi = 1;
my $ew_distance_mi = 1;
my $ns_distance_km = 1;
my $ew_distance_km = 1;
my $hashr;
my $i;
my $rCompliant;
my $rTotal;
my $sTotal;
my $percentTotal;
```

```
my $percentFGDC;
my $percentSuggested;
my $namesPrinted = "NONE";

# Variable to determine the number of occurrences of suggested and required
# FGDC elements
my $sCount = 0;
my $sCountFound = 0;
my $sCountTotal = 0;
my $sCountFoundTotal = 0;
my $sCompliant;
my $rCount = 0;
my $rCountFound = 0;
my $rCountTotal = 0;
my $rCountFoundTotal = 0;

# Color values to be filled in the report table
my $Pass = "Lime";
my $Failure = "Red";
my $Warning = "Yellow";

# hash representing the name of the item in %metadata and the path to that item in the tree
my %SearchList =
(

# FGDC Required Elements
    "r01_Data_Set_Title" => "idinfo_citation_citeinfo_title",
    "r02_Publication_Date" => "idinfo_citation_citeinfo_pubdate",
    "r03_Language" => "English",
    "r04_Data_Theme" => "idinfo_keywords_theme_themekey_0",
    "r05_Abstract" => "idinfo_descript_abstract",
    "r06_Metadata_POC" => "metainfo_metc_cntinfo_cntorgp_cntper",
    "r07_Metadata_Date" => "metainfo_metrd",

# FGDC Suggested Elements
      "s00_Spatial_Resolution" => "dataqual_posacc_horizpa_qhorizpa_horizpav",
    "s01_Distribution_Format" => "distinfo_resdesc",
    "s02_Additional_Spatial" => "dataqual_posacc_vertacc_vertaccr",
```

```
    "s03_Spatial_Representation" => "idinfo_citation_citeinfo_geoform",
    "s04_Reference_System" => "spref_horizsys_planar_gridsys_gridsysn",
    "s05_Lineage_Statement" => "dataqual_lineage_procstep_0_procdesc",
    "s06_Online_Resource" => "idinfo_citation_citeinfo_onlink",
    "s07_Metadata_Field" => "metainfo_metextns_0_onlink",
    "s08_Metadata_Standard_Name" => "metainfo_metstdn",
    "s09_Metadata_Standard_Version" => "metainfo_metstdv",
    "s10_Metadata_Language" => "English",
    "s11_Metadata_Character_Set" => "metainfo_metextns_0_metprof",
    "Geographic_Location" => "idinfo_spdom_bounding_westbc",
    "long2_s" => "idinfo_spdom_bounding_eastbc",
    "lat1_s" => "idinfo_spdom_bounding_northbc",
    "lat2_s" => "idinfo_spdom_bounding_southbc",
      "s14_Responsible_Party" => "idinfo_ptcontac_cntinfo_cntorgp_cntper",
      "s15_Data_Set_Character_Set" => "UFT-8",

#      Other non-required elements
    "n03_updatefrequency" => "idinfo_status_update",
    "n04_placekey" => "idinfo_keywords_place_placekey_1",
    "n05_geoid" => "spref_horizsys_geodetic_horizdn",
    "n06_ellipsoid" => "spref_horizsys_geodetic_ellips",
    "n07_semimajor" => "spref_horizsys_geodetic_semiaxis",
    "n08_flattening" => "spref_horizsys_geodetic_denflat",
    "n09_sdorganization" => "spdoinfo_direct",
    "n10_sdtstype" => "spdoinfo_ptvctinf_sdtsterm_sdtstype",
    "n11_objectcount" => "spdoinfo_ptvctinf_sdtsterm_ptvctcnt",
    "n12_attdefsystem" => "spref_horizsys_geodetic_ellips",
    "n13_contactorganization" => "idinfo_ptcontac_cntinfo_cntorgp_cntorg",
    "n14_mdorganization" => "metainfo_metc_cntinfo_cntorgp_cntorg",
    "n15_contactposition" => "idinfo_ptcontac_cntinfo_cntpos",
    "n16_mdposition" => "metainfo_metc_cntinfo_cntpos",
    "n21_useconstraints" => "idinfo_accconst",
    "n22_accessconstraints" => "idinfo_useconst",

# Attributes
    "attributes" => "eainfo_detailed_attr",
);
```

```
# Get the month and date in different forms
my ($month, $day, $year) = UnixDate("today", "%f", "%d", "%Y");
my $currentDate = UnixDate("today", "%Q");
my $dateString = UnixDate("today", "%F %r");

#Print version and date of program execution
print $version;
print "Run on $currentDate & $dateString\n";

$options = GetOptions
(
      "sourceDir=s" => \$sourceDir,
      "output=s" => \$output,
      "report=s" => \$report,
      "transaction=s" => \$transaction,

);

# Specify all XML files within the source directory
@files = <$sourceDir/*.xml>;
if (scalar @files == 0)
{

      print "No XML files found in directory: $sourceDir\n";
      exit;
}

print "\nNo output file specified.  Using ", $output, "\n" if($output eq './test.csv');

# If the file exists, open it for appending.  Otherwise, open it for writing
if (-e $output)
{
      unless (open OUTPUT_FILE, ">>$output")
      {
            die "Can't open the output file '$output': $!\n";
      }

      $foundfile = 1;
```

```
}
else
{
    unless (open OUTPUT_FILE, ">$output")
      {
            die "Can't open the output file '$output': $!\n";
      }

      print OUTPUT_FILE "File_Name,Current_Date";

}

# Open the report file for HTML reports for FGDC compliancy

print "\nNo report file specified.  Using ", $report, "\n" if($report eq './report.html');

if(! open REPORT_FILE, ">$report")
{
      die "Can't open the output file '$output': $!\n";
}



# If the transaction file exists, open it for appending.  Otherwise, open it for writing
if (-e $transaction)
{
      unless (open TRANSACTION_FILE, ">>$transaction")
      {
            die "Can't open the output file '$transaction': $!\n";
      }

      $foundfile = 1;

}
else
{
    unless (open TRANSACTION_FILE, ">$transaction")
```

```
	{
		die "Can't open the output file '$transaction': $!\n";
	}

}




# Once the REPORT_FILE has been opned, it will writed HTML to this file.  This contains
# the header and and table headers.  Be sure to include the master.css file in this
# directory or change this source of the master.css file in the link href below
	print REPORT_FILE '<html><head><link href="master.css" rel="stylesheet" type="text/css"
/></head><body>';
	print REPORT_FILE '<center><b>FGDC Compliancy Report</b></center>';
	print REPORT_FILE '<center><table border = 1 width = 80%>';
	print REPORT_FILE '<th>File Name</th><th>Layer Name</th><th>Required FGDC
Features</th><th>Suggested FGDC Features</th><th>Missing Features</th>';


# Open the TRANSACTION_FILE so it can be created for the associative data mining

# Check to see if today's data already exists by checking the date of the file.  If the data
# already exists, it will tell user that data has already been taken for the day
#else #open it, take a peek at the last date, if its ok, open it for appending.

if ( &FileHasTodaysData($output, 1) )
{
	print STDERR "Data has already been collected today\n";
	exit();
}


# traverse all files
foreach $filename (@files)
{
```

```
      $filename =~ s/\\/\//g;  # change all forward slashes to back-slashed to allow for proper
navigation
      print "\n\n..... Decomposing ", basename($filename), "  ..........\n";

# Create structure to traverse XML schema.  Before going to the next value, however,
# we need to reset the hash value.
      $tree = XMLin($filename);
      $metadata{"rMissing"} = '';
      $metadata{"fileName"} = $filename;
      $metadata{"sMissing"} = '';

    foreach $key (sort keys %SearchList)
    {
#        print "$key $SearchList{$key}\n";
        $Item = &FindItem($SearchList{$key});
#       print "Item: $Item\n\n";
        $Item =~ s/\n//g;
        $metadata{$key} = $Item;

# If the file begins with R, this means that it is required.  It will go through
# these hash values and check to see if it has been populated with NOT FOUND.  If it
# is found, then it will count the number of find so it can be calculated later.
           if($key =~ /^r/)
           {

# Increment the total number of required features that have been found in addition to
# just those that within this record
                  $rCount++;
                  $rCountTotal++;
                  if($Item ne 'NOT FOUND')
                  {
# increase the number of required features that have been found for all records and
# within this particular record
                        $rCountFound++;
                        $rCountFoundTotal++;

                  }
                  else
```

```
                {
                        $metadata{"rMissing"} .= $key;
                }


            }

# The hash value will populate the number of required features that have been populated
      $metadata{"rCountFound"} = $rCountFound;
# Hash value will have the numbe of possible required features
      $metadata{"rCount"} = $rCount;

# If the file begins with S, this means that it is suggested. In addition, the geographic
# location will is suggested, but contains 4 elements.  This wil check 1 element.  It will go through
# these hash values and check to see if it has been populated with NOT FOUND.  If it
# is found, then it will count the number of find so it can be calculated later.

            if($key =~ /^s|Geographic_Location/)
            {

# Increment the total number of suggested features that have been found in addition to
# just those that within this record
                $sCount++;
                $sCountTotal++;
                if($Item ne 'NOT FOUND')
                {
# increase the number of suggested features that have been found for all records and
# within this particular record
                        $sCountFound++;
                        $sCountFoundTotal++;
                }
                else
                {
                        $metadata{"sMissing"} .= $key;
                }


            }
```

```
# The hash value will populate the number of suggested features that have been populated
      $metadata{"sCountFound"} = $sCountFound;
# Hash value will have the numbe of possible suggested features
      $metadata{"sCount"} = $sCount;


    }

# Resent the counts for the suggested and required for the next record record.
      $rCount = 0;
      $rCountFound = 0;
      $sCount = 0;
      $sCountFound = 0;

# Compute the latitude and longitude from the bounding coordinates
    $metadata{"s12_locationlong"} = ($metadata{"Geographic_Location"} + $metadata{"long2_s"}) / 2;
    $metadata{"s13_locationlat"} = ($metadata{"lat1_s"} + $metadata{"lat2_s"}) / 2;
      $metadata{"s00_Spatial_Resolution"} =~ s/\+..//;      # replace + sign with spaces
      $metadata{"s00_Spatial_Resolution"} =~ s/meters//;      # replace meters with spaces, leaving
just a number

#     Non-Required Elements.  Use for Data Mining Purposes
#     Call up GIS perl functionality to perform geographic computations
      my $geo = Geo::Ellipsoid->new(ellipsoid => 'WGS84', units => 'degrees');

#Compute the area of the extent using the Geo Perl function
      $ns_distance_mi = $geo->range($metadata{"lat1_s"}, $metadata{"Geographic_Location"},
$metadata{"lat2_s"}, $metadata{"Geographic_Location"})/ $meters_in_a_mile;
      $ew_distance_mi = $geo->range($metadata{"s13_locationlat"}, $metadata{"Geographic_Location"} ,
$metadata{"s13_locationlat"}, $metadata{"long2_s"})/ $meters_in_a_mile;
      $ns_distance_km = $ns_distance_mi / $miles_in_a_kilometer;
      $ew_distance_km = $ew_distance_mi / $miles_in_a_kilometer;
      $metadata{"n17_xdistance_mi"} = $ew_distance_mi;
      $metadata{"n18_ydistance_mi"} = $ns_distance_mi;
      $metadata{"n19_xdistance_km"} = $ew_distance_km;
      $metadata{"n20_ydistance_km"} = $ns_distance_km;
```

```
#     print "\n NS Distance is $ns_distance_mi";
#     print "\n EW Distance is $ew_distance_mi";
#     print "\n NS Distance is $ns_distance_km";
#     print "\n EW Distance is $ew_distance_km";

#     Calculate the area of extent based on values derived from range function above
      $metadata{"n01_areaofextentsqmi"} = $ns_distance_mi * $ew_distance_mi;
      $metadata{"n02_areaofextentsqkm"} = $ns_distance_km * $ew_distance_km;

#
# Get the number of attributes from the metadata

# print "\n\nAttribute Table Total: ",scalar
@{$metadata{"attributes"}},"\nIndex\tLabel\t\tDefinition\n";
$i = 0;
#foreach $hashr (@{$metadata{"attributes"}})
#{
#    if(length(${$hashr}{"attrlabl"}) <8)
#    {
#        print $i,"\t",${$hashr}{"attrlabl"},"\t\t",${$hashr}{"attrdef"},"\n";
#    }
#    else
#    {
#        print $i,"\t",${$hashr}{"attrlabl"},"\t",${$hashr}{"attrdef"},"\n";
#    }
#    $i++;
#}

#$metadata{"n23_numattributes"} = scalar @{$metadata{"attributes"}};
print "\n";

# Add the other file headings to first line based on found flag value set above
      unless ($foundfile)
      {
            foreach $key (sort keys %metadata)
            {

                  print OUTPUT_FILE ",$key";
```

```
            }

            $foundfile = 1;
            print OUTPUT_FILE "\n";
      }

      print "\n..... Metadata Found for ", basename($filename), "  ...........\n";
      print OUTPUT_FILE basename($filename),",", $currentDate;

      # Print out the transaction Number

      foreach $key (sort keys %metadata)
      {
            $_ = $metadata{$key};
            s/,/ /g;
            $metadata{$key} = $_;
            print "$key => $metadata{$key}\n";
            print OUTPUT_FILE ",'$metadata{$key}'";

            #   Writing to HTML Report File
            if($key eq "fileName")
            {

# Write the table and find the appropriate hash element
                  print REPORT_FILE '<tr><td>',$metadata{"fileName"},'</td>';
                  print REPORT_FILE '<td>',$metadata{"r01_Data_Set_Title"},'</td>';

# Check to see if the required count is the same as the required found.  This means
# that all values are FGDC compliant.  Make this table element with a green background,
# otherwise make it with a red background.  Also increment to total number of compliant layers
                  if($metadata{"rCountFound"} eq $metadata{"rCount"})
                  {
                        print REPORT_FILE "<td bgcolor = $Pass>",'</td>';
                        $rCompliant++;

                  }
                  else
```

```
                  {
                        print REPORT_FILE "<td bgcolor =
$Failure><center>",$metadata{"rCountFound"},'</center></td>';

                  }

# Check to see if the suggested count is the same as the suggested found.  This means
# that all values are FGDC compliant.  Make this table element with a green background,
# otherwise make it with a yellow background.  Also increment to total number of compliant layers
                  if($metadata{"sCountFound"} eq $metadata{"sCount"})
                  {
                        print REPORT_FILE "<td bgcolor = $Pass>",'</td>';
                        $sCompliant++;

                  }
                  else
                  {
                        print REPORT_FILE "<td bgcolor =
$Warning><center>",$metadata{"sCountFound"},'</center></td>';

                  }
# Populate the last table element with the elements.  These are found in the
# hash elements that were created when the elements were checked.
                  if(($metadata{"rMissing"} ne '') & ($metadata{"sMissing"} ne ''))
                  {
                        $namesPrinted = join(", ",$metadata{"rMissing"} , $metadata{"sMissing"});
                  }
                  else
                  {
                        $namesPrinted = join(", ",$metadata{"rMissing"} , $metadata{"sMissing"});

                  }

# Perl regular expression to replace underspace with spaces and the hash names
# beginning with r and w with nothing
      #           $namesPrinted =~ s/r.._//;
      #           $namesPrinted =~ s/s.._//;
      #           $namesPrinted =~ s/_/ /g;
```

```
                  if($namesPrinted eq '')
                  {
                        $namesPrinted = 'NONE';
                  }

                  print REPORT_FILE '<td>',$namesPrinted ,'</td></tr>';


# Total number of required and suggested elements are incremented
                  $rTotal++;
                  $sTotal++;
            }

      }

      print OUTPUT_FILE "\n";


      # Print values to the transaction file before ending this loop
      # this will go through look at certain keys and populate the transaction
      # file the appropriate value

# Increase transaction number by 1.  All values for the same record will have
# the same transaction number
      $transaction_number = $transaction_number + 1;

# Find the area of the coveragea and put it into one of 3 different values

      if($metadata{"n01_areaofextentsqmi"} < 5)
      {
            print TRANSACTION_FILE $transaction_number,"\t","Area_of_Layer=Low","\n";
      }
      if($metadata{"n01_areaofextentsqmi"} >= 5 && $metadata{"n01_areaofextentsqmi"} <= 100)
      {
            print TRANSACTION_FILE $transaction_number,"\t","Area_of_Layer=Medium","\n";
      }
      if($metadata{"n01_areaofextentsqmi"} > 100)
```

```
	{
		print TRANSACTION_FILE $transaction_number,"\t","Area_of_Layer=High","\n";
	}

# Find the update frequency and replace the spaces with an underscore
	my $update_trans;
	$update_trans = $metadata{"n03_updatefrequency"};
	$update_trans =~ s/ /_/g;
	print TRANSACTION_FILE $transaction_number,"\t","Update_Frequency=".$update_trans,"\n";


# Find the place key and replace the spaces with an underscore
	my $placekey_trans;
	$placekey_trans = $metadata{"n04_placekey"};
	$placekey_trans =~ s/ /_/g;
	print TRANSACTION_FILE $transaction_number,"\t","Place_Key=".$placekey_trans,"\n";


# Find the update geoid and replace the spaces with an underscore
	my $geoid_trans;
	$geoid_trans = $metadata{"n05_geoid"};
	$geoid_trans =~ s/ /_/g;
	print TRANSACTION_FILE $transaction_number,"\t","Geoid=".$geoid_trans,"\n";

# Find the update ellipsoid and replace the spaces with an underscore
	my $ellipsoid_trans;
	$ellipsoid_trans = $metadata{"n06_ellipsoid"};
	$ellipsoid_trans =~ s/ /_/g;
	print TRANSACTION_FILE $transaction_number,"\t","Ellipsoid=".$ellipsoid_trans,"\n";


# Find object county and put it into one of 3 different values
	if($metadata{"n11_objectcount"} <= 10)
	{
		print TRANSACTION_FILE $transaction_number,"\t","Object_Count=Low","\n";
	}
	if($metadata{"n11_objectcount"} >= 11 && $metadata{"n11_objectcount"} <= 100)
	{
```

```
            print TRANSACTION_FILE $transaction_number,"\t","Object_Count=Medium","\n";

      }
      if($metadata{"n11_objectcount"} > 100)
      {
            print TRANSACTION_FILE $transaction_number,"\t","Object_Count=High","\n";
      }

# Find the contact organization and replace the spaces with an underscore
      my $contact_org_trans;
      $contact_org_trans = $metadata{"n13_contactorganization"};
      $contact_org_trans =~ s/ /_/g;
      print TRANSACTION_FILE
$transaction_number,"\t","Contact_Organization=".$contact_org_trans,"\n";

# Find the metadata organization and replace the spaces with an underscore
      my $metadata_org_trans;
      $metadata_org_trans = $metadata{"n14_mdorganization"};
      $metadata_org_trans =~ s/ /_/g;
      print TRANSACTION_FILE
$transaction_number,"\t","Metadata_Organization=".$metadata_org_trans,"\n";

# Find the contact position and replace the spaces with an underscore
      my $contact_position_trans;
      $contact_position_trans = $metadata{"n15_contactposition"};
      $contact_position_trans =~ s/ /_/g;
      print TRANSACTION_FILE
$transaction_number,"\t","Contact_Position=".$contact_position_trans,"\n";

# Find the metadata position and replace the spaces with an underscore
      my $metadata_pos_trans;
      $metadata_pos_trans = $metadata{"n16_mdposition"};
      $metadata_pos_trans =~ s/ /_/g;
      print TRANSACTION_FILE
$transaction_number,"\t","Metadata_Organization=".$metadata_pos_trans,"\n";

# Find the use constraints and replace the spaces with an underscore
      my $use_contraints_trans;
```

```
	$use_contraints_trans = $metadata{"n21_useconstraints"};
	$use_contraints_trans =~ s/ /_/g;
	print TRANSACTION_FILE $transaction_number,"\t","Use_Constraints=".$use_contraints_trans,"\n";

# Find the access constraints and replace the spaces with an underscore
	my $access_contraints_trans;
	$access_contraints_trans = $metadata{"n22_accessconstraints"};
	$access_contraints_trans =~ s/ /_/g;
	print TRANSACTION_FILE
$transaction_number,"\t","Access_Constraints=".$access_contraints_trans,"\n";

# Find number of attributes and put it into one of 3 different values
	if($metadata{"n23_numattributes"} <= 10)
	{
		print TRANSACTION_FILE $transaction_number,"\t","Number_Attributes=Low","\n";
	}
	if($metadata{"n23_numattributes"} >= 11 && $metadata{"n23_numattributes"} <= 50)
	{
		print TRANSACTION_FILE $transaction_number,"\t","Number_Attributes=Medium","\n";
	}
	if($metadata{"n23_numattributes"} > 50)
	{
		print TRANSACTION_FILE $transaction_number,"\t","Number_Attributes=High","\n";
	}

# Find publication date and put it into one of 3 different values
	if(($metadata{"r02_Publication_Date"} eq 'unknown') || ($metadata{"r02_Publication_Date"} eq
'NOT FOUND'))
	{
		print TRANSACTION_FILE $transaction_number,"\t","Publication_Date=Unknown","\n";
	}
	if($metadata{"r02_Publication_Date"} >= 20070101)
	{
		print TRANSACTION_FILE $transaction_number,"\t","Publication_Date=New","\n";
	}
	if($metadata{"r02_Publication_Date"} >= 20040101 && $metadata{"r02_Publication_Date"} <=
20061231)
	{
```

```
            print TRANSACTION_FILE $transaction_number,"\t","Publication_Date=Medium","\n";
      }
      if($metadata{"r02_Publication_Date"} < 20040101)
      {
            print TRANSACTION_FILE $transaction_number,"\t","Publication_Date=Old","\n";
      }

# Find the data theme and replace the spaces with an underscore
      my $data_theme_trans;
      $data_theme_trans = $metadata{"r04_Data_Theme"};
      $data_theme_trans =~ s/ /_/g;
      print TRANSACTION_FILE $transaction_number,"\t","Data_Theme=".$data_theme_trans,"\n";


# Find the metadata POC and replace the spaces with an underscore
      my $metadata_poc_trans;
      $metadata_poc_trans = $metadata{"r06_Metadata_POC"};
      $metadata_poc_trans =~ s/ /_/g;
      print TRANSACTION_FILE $transaction_number,"\t","Metadata_POC=".$metadata_poc_trans,"\n";


# Find metadata date and put it into one of 3 different values
my $found_metadata = 0;
      if(($metadata{"r07_Metadata_Date"} eq 'unknown') || ($metadata{"r07_Metadata_Date"} eq 'NOT
FOUND'))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Metadata_Date=Unknown","\n";
            $found_metadata = 1;
      }
      if(($metadata{"r07_Metadata_Date"} >= 20070101) && ($found_metadata == 0))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Metadata_Date=New","\n";
      }
      if(($metadata{"r07_Metadata_Date"} >= 20040101 && $metadata{"r07_Metadata_Date"} <= 20061231)
&& ($found_metadata == 0))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Metadata_Date=Medium","\n";
      }
```

```
      if(($metadata{"r07_Metadata_Date"} < 20040101) && ($found_metadata == 0))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Metadata_Date=Old","\n";
      }


# Find spatial resolution and put it into one of 3 different values
my $found_horizontal = 0;

      if(($metadata{"s00_Spatial_Resolution"} eq 'unknown') || ($metadata{"s00_Spatial_Resolution"}
eq 'NOT FOUND'))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Horizontal_Accuracy=Unknown","\n";

            $found_horizontal = 1;
      }
      if(($metadata{"s00_Spatial_Resolution"} <= 50) && ($found_horizontal == 0))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Horizontal_Accuracy=Excellent","\n";
      }
      if(($metadata{"s00_Spatial_Resolution"} > 50) && ($metadata{"s00_Spatial_Resolution"} <= 100)
&& ($found_horizontal == 0))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Horizontal_Accuracy=Medium","\n";
      }
      if(($metadata{"s00_Spatial_Resolution"} > 100) && ($found_horizontal == 0))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Horizontal_Accuracy=Poor","\n";
      }

# Find location from the center coordinates and put it into one of 6 different quadrants
# These zones are aboslute locations based on latitude and longitude.  The different
# regions are Northeast, Southeast, Upper Midwest, Lower Midwest, Northwest and Southwest
      if(($metadata{"s12_locationlong"} < -104) && ($metadata{"s13_locationlat"} > 39))
      {
            print TRANSACTION_FILE $transaction_number,"\t","Location=Northwest","\n";
      }
      if(($metadata{"s12_locationlong"} < -104) && ($metadata{"s13_locationlat"} < 39))
```

```
	{
		print TRANSACTION_FILE $transaction_number,"\t","Location=Southwest","\n";
	}
	if(($metadata{"s12_locationlong"} > -87) && ($metadata{"s13_locationlat"} > 39))
	{
		print TRANSACTION_FILE $transaction_number,"\t","Location=Northeast","\n";
	}
	if(($metadata{"s12_locationlong"} > -87) && ($metadata{"s13_locationlat"} < 39))
	{
		print TRANSACTION_FILE $transaction_number,"\t","Location=Northeast","\n";
	}
	if((($metadata{"s12_locationlong"} < -87) && ($metadata{"s12_locationlong"} > -104)) &&
($metadata{"s13_locationlat"} > 39))
	{
		print TRANSACTION_FILE $transaction_number,"\t","Location=Upper_Midwest","\n";
	}
	if((($metadata{"s12_locationlong"} < -87) && ($metadata{"s12_locationlong"} > -104)) &&
($metadata{"s13_locationlat"} < 39))
	{
		print TRANSACTION_FILE $transaction_number,"\t","Location=Lower_Midwest","\n";
	}

# Find the responsible part and replace the spaces with an underscore
	my $responsible_party_trans;
	$responsible_party_trans = $metadata{"s14_Responsible_Party"};
	$responsible_party_trans =~ s/ /_/g;
	print TRANSACTION_FILE
$transaction_number,"\t","Responsible_Party=".$responsible_party_trans,"\n";


# End of writing to the transaction file

}


# Output at the end of table and further calculations
$percentTotal = sprintf("%.2f",100 * $rCompliant / $rTotal);
$percentFGDC = sprintf("%.2f",100 * $rCountFoundTotal / $rCountTotal);
```

```
# Finish writing the table and calculate the number of compliant-required layers.
print REPORT_FILE '</table><br><center>',$rCompliant, ' out of ',$rTotal, ' layers (', $percentTotal
,'%) had all of the FGDC Required metadata components';
# Calculate and print the total number of required elements that were populated in all layers
print REPORT_FILE '<br><center>',$rCountFoundTotal, ' out of ',$rCountTotal, ' individual FGDC
required elements (', $percentFGDC ,'%) were adequately populated<br><hr width = 60% color = Green
align = CENTER>';

# FGDC suggested element calculations.
$percentTotal = sprintf("%.2f",100 * $sCompliant / $sTotal);
$percentSuggested = sprintf("%.2f",100 * $sCountFoundTotal / $sCountTotal);

# Print out results from total number of suggested elements

print REPORT_FILE '</table><br><center>',$sCompliant, ' out of ',$sTotal, ' layers (', $percentTotal
,'%) had all of the FGDC Suggested metadata components';
print REPORT_FILE '</table><br><center>',$sCountFoundTotal, ' out of ',$sCountTotal, ' individual
FGDC required elements (', $percentSuggested ,'%) were adequately populated<br><hr width = 60% color
= Green align = CENTER>';

# Finish writing HTML file and close both the CSV and HTML file
print REPORT_FILE '</body></html>';
close OUTPUT_FILE;
close REPORT_FILE;
close TRANSACTION_FILE;



exit;

# Routine takes in current file as parameter to check if data has been collected for the day
sub FileHasTodaysData ($)
{
      my $csv_file = $_[0];
      my $replace = $_[1];
    my $not_replaced = 0;
```

```
  my ($fh, $file) = tempfile();

  unless (open TEMP_OUTPUT_FILE, "<$csv_file")
  {
              print STDERR "Can't open the csv_file file '$csv_file': $!\n";
              close TEMP_OUTPUT_FILE;
              return 0;
  }

  my $currentDate = UnixDate("today", "%Q");
print "Checking if $currentDate already in $csv_file\n";

  #if there is data from today delete it, by omiting it from temp file
  while (<TEMP_OUTPUT_FILE>)
  {
        unless (/$currentDate/)
        {
              print $fh $_;
        }
        else
        {
        if($replace)
        {
                  print STDOUT "Dropping $_\n";
        }
        else
        {
                  print $fh $_;
                  print STDOUT "Keeping $_\n";
            $not_replaced = 1;
        }
        }
  }

  close TEMP_OUTPUT_FILE;
  close $fh;

  # make the temp file the new file
```

```
      copy($file, $csv_file) or
            warn "Couldn't rename $file to $csv_file: $!\n";

      return $not_replaced;
}


# subroutine to find the elemnet at the end of the path
# SearchList is elemnet path separated by "_"
# returns the vale of the item
sub FindItem
{
    my $SearchList = $_[0];
    my @SearchListPath;
    my $key;
    my $ref;
    my $Item;
    my $Index;

    @SearchListPath = split("_",$SearchList);
    print "Path: ",join("->",@SearchListPath),"\n";
    $ref = \%$tree;

      if (scalar @SearchListPath == 1)

      {
            return $SearchList;
      }

    while( scalar @SearchListPath > 0)
    {
        print "SearchPath: ", join(" ",@SearchListPath);
        $key = shift(@SearchListPath);
            print " Key: $key\n";
        $Item = $key;
        last unless ( exists $$ref{$key});
        print "$key => $$ref{$key} ",ref($$ref{$key}),"\n";
        if(ref($$ref{$key}) eq "HASH")
```

```
        {
            $ref = $$ref{$key};
        }
        elsif (ref($$ref{$key}) eq "ARRAY")
        {
            $ref = $$ref{$key};
            if(scalar @SearchListPath == 0)
            {
               $Item = $ref;
                last;

            }
            $Index = shift(@SearchListPath);
            $Item = $$ref[$Index];
#            print "Array[$Index]: $Item\n";
            if(ref($Item) eq "HASH")
            {
                $ref = $Item;
                next;
            }
            last ;
        }
        else
        {
            $Item = $$ref{$key};
        }
    }

      $Item = "NOT FOUND" unless ( $Item ne $key);
      print "Item Found: ",$Item, "\n";
    $Item;
}
```

APPENDIX D: R CODE USE TO CREATE HISTOGRAMS FOR HORIZONTAL AND TEMPORAL ACCURACY

```
# function to convert month, day, year list to year.xxx format
# Input:
#     data - list object with $Month $Day and $Year components (all other components of data are
ignored
#
# Output: vector of dates in form year.xxx of same length as input data where xxx is n/365 days
get_date<- function(data)
{
    month.length<-c(31,28,31,30,31,30,31,31,30,31,30,31)     # lengths of every month
    t<-NULL                                                   # start off with NULL value
    i<-0                                                      # starting value
    while(i < length(data$Month))                             # check entire input list
    {
        m <- data$Month[i+1]                                  # get month
        d <- data$Day[i+1]                                    # get day
        y <- data$Year[i+1]                                   # get year

        if(m == 1)                                            # check if jan
        {
            d1<- d                                            # yes so days are total
        }
        else
        {
            s<-cumsum(month.length[1:(m-1)])                  # no so get total days of all months up
to this one
            d1<- d + s[length(s)]                             # add in remaing days
        }
#        cat(m,d,y,(y + (d1/365)),"\n")
        t<- c(t,(y + (d1/365)))                               # calculate the next value and add to
vector
        i<- i+1                                               # go to next element in input list
    }
    return(t)                                                 # return the result vector
}
```

```
test_convert<-function(InFilename, device = "jpg")  # This function will convert to a
{

# Varible Declaration


# get the data

    ydata<-read.csv(InFilename)
#     dateSTR<-substr(as.character(ydata$r07_Metadata_Date),2,9)  # Parse year, month and day from
the string

# Replace the quotations with nothing

      ydata$dateParse<-gsub("'", "" ,as.character(ydata$r02_Publication_Date))
      ydata$NumChars<-nchar(as.character(ydata$dateParse))
      ydata_good<-list(date=ydata$dateParse[ydata$NumChars == 8])
      ydata_good$Month<-as.integer(substr(ydata_good$date,5,6))
      ydata_good$Year<-as.integer(substr(ydata_good$date,1,4))
      ydata_good$Day<-as.integer(substr(ydata_good$date,7,8))

   ydata_good$theDate<-get_date(ydata_good)           # pass values to date
   temporal_mean<-mean(ydata_good$theDate)
   temporal_median<-median(ydata_good$theDate)
   temporal_variance<-var(ydata_good$theDate)
   temporal_std<-sd(ydata_good$theDate)

    ydata_good$std_lag<-(ydata_good$theDate - temporal_mean) / temporal_std

    leg = c("Temporal Mean", "Temporal Median", "Critical Theshold")               # Create the
legend

        switch(device,                                          # check the device and set driver
accordingly
            x11 =  x11(),
            pdf =  pdf(OutFilenameMod,width=10,height=8),
```

```
            png =  bitmap(OutFilenameMod,width=1280,height=1024,units="px"),
            jpg = jpeg("output_temporal.jpg", width = 800, height = 600, quality = 95),
            x11())                                          # default is x11

# Calculate the Number of Breaks
      numBreaks = 2 * ceiling(diff(range(ydata_good$theDate)))

    hist_x<-hist(ydata_good$theDate, plot = FALSE, breaks = numBreaks)        # Creates the
statistical information without needing to draw it on the plot

#     return(ydata_good)



    color_bars<-rep("GREEN", sum(hist_x$counts))            # Colors all histograms to be green
    threshold<-temporal_mean - 1.5 * temporal_std           # Danger threshold more than 1.5*std -
could be changed to quintiles in future
    threshold_yellow<-temporal_mean - .5 * temporal_std     # Yellow threshold - Between .5 and 1.5
below mean - could be changed to quintiles

    color_bars[hist_x$breaks < threshold_yellow]<-"YELLOW"  # overwrites all greens which satsify
criteria
    color_bars[hist_x$breaks < threshold]<-"RED"      # overwrites the reds which satisfy the
threshold
    hist(ydata_good$theDate, main = "Temporal Assessment", xlab = "Year", ylab = "Frequency", col =
color_bars, breaks = numBreaks,ylim = c(0, 1.2 * max(hist_x$counts)))     # Creates the histogram
which can be displayed
    lines(c(temporal_mean, temporal_mean),c(0,max(hist_x$counts) * 1.1), lty = 2, lwd = 3, col =
"BLACK")
    text_offset<- .25 * max(diff(hist_x$breaks))
    text(x = temporal_mean - text_offset , y = .25 * max(hist_x$counts), labels =
get_date_text(temporal_mean), srt = 90, col = "BLACK", cex = .75)
    lines(c(temporal_median, temporal_median),c(0,max(hist_x$counts) * 1.1), lty = 2, lwd = 3, col =
"BLUE")
    text(x = temporal_median - text_offset , y = .55 * max(hist_x$counts), labels =
get_date_text(temporal_median), srt = 90, col = "BLUE", cex = .75)
    lines(c(threshold, threshold),c(0,max(hist_x$counts) * 1.1), lty = 2, lwd = 5, col = "RED3")
    legend(x = "topleft", legend = leg, lty = 2, , lwd = 2, col = c("BLACK","BLUE","RED3"))
```

```
    t<-paste("Created on: ", date(), "\n Number of Samples:", sum(hist_x$counts), "\n Number of
Missing Records:", nrow(ydata) - length(ydata_good$theDate))
    text(x = mean(hist_x$breaks) + .7 *(max(hist_x$breaks) - mean(hist_x$breaks)), y = 1.1 *
max(hist_x$counts) , labels = t, cex = .5)

    dev.off()

    sink("publication.html")   # Write data to HTML file
      cat("<html><head><link href='master.css' rel='stylesheet'
type='text/css'></head><body><TABLE><TD>")
      cat("<IMG SRC = 'output_temporal.jpg'>")
      cat("</TD><TD><B>Temporal Mean: </B>", get_date_text(temporal_mean), "<BR>")
      cat("<B>Temporal Median: </B>", get_date_text(temporal_median),"<BR>")
      cat("<B>Temporal Variance: </B>", get_time_text(temporal_variance), "<BR>")
      cat("<B>Temporal Standard Dev.: </B>", get_time_text(temporal_std), "<BR>")
      cat("</HTML>")
    sink()        # Turn off writing data to HTML file

    ydata_list<-list(data = ydata, mean = temporal_mean, median = temporal_median, variance =
temporal_variance, std = temporal_std)



# Extract the good horizontal accuracy values
      ydata$accuracyParse<-gsub("'", "" ,as.character(ydata$s00_Spatial_Resolution))
      options(warn= -1)
      ydata_accuracy<-list(horizontalAccuracy=as.numeric(ydata$accuracyParse))
      options(warn= 0)
      ydata_accuracy$horizontalAccuracy<-
ydata_accuracy$horizontalAccuracy[!is.na(ydata_accuracy$horizontalAccuracy)]
#     return(ydata_accuracy)

    accuracy_mean<-mean(as.double(ydata_accuracy$horizontalAccuracy))
    accuracy_median<-median(as.double(ydata_accuracy$horizontalAccuracy))
    accuracy_variance<-var(as.double(ydata_accuracy$horizontalAccuracy))
    accuracy_std<-sd(as.double(ydata_accuracy$horizontalAccuracy))
```

```
    ydata_list<-c(ydata_list, acc_mean = accuracy_mean, acc_median = accuracy_median, acc_var =
accuracy_variance, acc_std = accuracy_std)

    switch(device,                                          # check the device and set driver
accordingly
            x11 =  x11(),
            pdf =  pdf(OutFilenameMod,width=10,height=8),
            png =  bitmap(OutFilenameMod,width=1280,height=1024,units="px"),
            jpg = jpeg("output_accuracy.jpg", width = 800, height = 600, quality = 95),
            x11())                                            # default is x11



    breaks_acc<-seq(0,max(as.double(ydata_accuracy$horizontalAccuracy)) + 12.7 ,by = 12.7)
    cat(breaks_acc, "\n")
      hist_acc<-hist(as.double(ydata_accuracy$horizontalAccuracy), plot = FALSE, main = "Accuracy
Assessment", breaks = breaks_acc , xlab = "Horizontal Accuracy (Meters)", ylab = "Frequency", ylim =
c(0, 1.2 * max(hist_acc$counts)))     # Creates the histogram which can be displayed
    hist(as.double(ydata_accuracy$horizontalAccuracy), main = "Accuracy Assessment", breaks =
breaks_acc , xlab = "Horizontal Accuracy (Meters)", ylab = "Frequency", ylim = c(0, 1.2 *
max(hist_acc$counts)))     # Creates the histogram which can be displayed

#      Draw the appropriate lines for the different thresholds
    lines(c(12.7, 12.7),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3, col = "BLUE")
    lines(c(25.4, 25.4),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3, col = "RED3")
    lines(c(50.8, 50.8),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3, col = "GREEN2")
    lines(c(101.6, 101.6),c(0, max(hist_acc$counts)* 1.1), lty = 2, lwd = 3, col = "MAGENTA")

    leg_accuracy = c("1:25,000", "1:50,000", "1:100,000", "1:250,000")              #      Create the
legend
    legend(x = "topright", title = "Mapping Thresholds", legend = leg_accuracy, lty = 2, , lwd = 2,
col = c("BLUE","RED3","GREEN2","MAGENTA"), cex = .6, bg = "WHITE")

    t<-paste("Created on: ", date(), "\n Number of Samples:", sum(hist_acc$counts), "\nNumber of
Dropped Records :", nrow(ydata) - length(ydata_accuracy$horizontalAccuracy))
    text(x = hist_acc$breaks[3], y = 1.15 * max(hist_acc$counts) , labels = t, cex = .5)
```

```
    cat(hist_acc$intensities, "\n")

   dev.off()

    sink("horizontal.html")

      cat("<html><head><link href='master.css' rel='stylesheet'
type='text/css'></head><body><TABLE><TD>")
      cat("<IMG SRC = 'output_accuracy.jpg'>")
    cat("</TD><TD><B>Mean: </B>", accuracy_mean, " Meters<BR>")
      cat("<B>Median: </B>", accuracy_median," Meters<BR>")
      cat("<B>Variance: </B>", accuracy_variance, " Meters<BR>")
      cat("<B>Standard Deviation: </B>", accuracy_std, " Meters<BR>")
      cat("</HTML>")
    sink()


}


# function to get a text string of the form mm/dd/yy
# Input:
#     date - in format year.xxx where xxx is fraction of 365
#
# Output: text string of format "mm/dd/yy"
get_date_text<- function(date)
{

    theYear<-ceiling(date)

    if(floor(theYear / 4) == (theYear / 4))
     {
       febDays = 29
       daysYear = 366

      }
      else
      {
```

```
            febDays = 28
            daysYear = 365
      }

    month.length<-c(31,febDays,31,30,31,30,31,31,30,31,30,31)     # lengths of every month
    month.sum<-cumsum(month.length)                              # get a running sum of all the months
    dt<- round((date - floor(date)) * daysYear)                       # get the day in the year
    d<-dt                                                        # save it
    rd<-month.sum - d                                            # generate a vector of remaing days in
the year
    m<-1                                                         # assume month is 1
    while(rd[m] < 0)                                             # check if you have reached the desired
month
    {
        d<- dt - month.sum[m]                                    # no so get remaing days from running
sum
        m<-m+1                                                   # point to next month
    }
    y=floor(date)                                          # get last digits of year
    dt<-paste(sprintf("%02d",m),"/",sprintf("%02d",d),"/",sprintf("%04d",y),sep="")
# make text string
    return(dt)                                                   # return the final text string
}


# function to get a text string of the form mm/dd/yy
# Input:
#     date - in format year.xxx where xxx is fraction of 365
#
# Output: text string of format Years, months and days
get_time_text<- function(date)
{

    month.length<-c(31,28,31,30,31,30,31,31,30,31,30,31)     # lengths of every month
    month.sum<-cumsum(month.length)                           # get a running sum of all the months
    dt<- round((date - floor(date)) * 365.24219)                   # get the day in the year
    d<-dt                                                     # save it
```

```
    rd<-month.sum - d                                       # generate a vector of remaing days in
the year
    m<-1                                                    # assume month is 1
    while(rd[m] < 0)                                        # check if you have reached the desired
month
    {
        d<- dt - month.sum[m]                               # no so get remaing days from running
sum
        m<-m+1                                              # point to next month
    }
    y=floor(date)                                       # get last digits of year
    dt<-paste(sprintf("%d",y)," years ",sprintf("%d",m)," months ",sprintf("%d",d)," days", sep="")
# make text string
    return(dt)                                              # return the final text string
}
```

APPENDIX E: PHP CODE USED TO BUILD DYNAMIC FORM ELEMENTS FROM CSV FILE

```
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title >Exploratory Data Analysis</title>
</head>
<body>

<?php
// Local PHP variables


$tableWidth = "900";
//$tableWidth = "100%";
//$bgColor = "#FFFFFF";
$bgColor = "white";
//$bgColorRequired = "#CCFFFCC";
$bgColorRequired = "palegreen";
//$tableWidthColumn1Required = "138";
$tableWidthColumn1Required = "20%";
//$tableWidthColumn2Required = "431";
$tableWidthColumn2Required = "40%";
//$tableWidthColumn3Required = "154";
$tableWidthColumn3Required = "20%";
//$tableWidthColumn4Required = "154";
$tableWidthColumn4Required = "20%";
$MySqlServerName = "localhost";
$MySqlDatabaseName = "eda_db";
$MySqlDatabaseSizes = array
    (
        "varchar(50)", // column A
        "varchar(15)", // column B
        "varchar(20)", // column C
        "varchar(30)", // column D
        "varchar(50)", // column E
        "varchar(30)", // column F
        "varchar(30)", // column G
```

```
"varchar(30)", // column H
"varchar(30)", // column I
"varchar(30)", // column J
"varchar(30)", // column K
"varchar(30)", // column L
"varchar(30)", // column M
"varchar(30)", // column N
"varchar(30)", // column O
"varchar(30)", // column P
"varchar(30)", // column Q
"varchar(30)", // column R
"varchar(30)", // column S
"varchar(30)", // column T
"varchar(30)", // column U
"varchar(40)", // column V
"varchar(30)", // column W
"varchar(30)", // column X
"varchar(30)", // column Y
"varchar(30)", // column Z
"varchar(50)", // column AA
"varchar(30)", // column AB
"varchar(60)", // column AC
"varchar(60)", // column AD
"varchar(50)", // column AE
"varchar(75)", // column AF
"varchar(30)", // column AG
"varchar(30)", // column AH
"varchar(30)", // column AI
"varchar(256)", // column AJ
"varchar(40)", // column AK
"varchar(30)", // column AL
"varchar(30)", // column AM
"varchar(30)", // column AN
"varchar(30)", // column AO
"varchar(30)", // column AP
"varchar(30)", // column AQ
"varchar(128)", // column AR
"varchar(30)", // column AS
```

```
        "varchar(30)", // column AT
        "varchar(300)", // column AU
        "varchar(60)", // column AV
        "varchar(60)", // column AW
        "varchar(60)", // column AX
        "varchar(30)", // column AY
        "varchar(30)", // column AZ
        "varchar(50)", // column BA
        "varchar(15)", // column BB
        "varchar(20)", // column BC
        "varchar(60)", // column BD
        "varchar(50)", // column BE
        "varchar(30)", // column BF
        "varchar(30)", // column BG
        "varchar(60)", // column BH
    );
$MySqlTableName = "eda_table";
$inputCSVFilename = "test.csv";
$expectedNumberOfFields = 60;
$csvField = range(0,$expectedNumberOfFields -1);
$numColumnsInForm = 4;

// connect to MySql server
$con = mysql_connect("$MySqlServerName","root","admin");
if (!$con)
{
    die('Could not connect: ' . mysql_error()) . "<br />";
}
else
{
    echo "Connected to MySql Server $MySqlServerName<br />";
}

// Drop  database
if (mysql_query("DROP DATABASE IF EXISTS $MySqlDatabaseName",$con))
{
    echo "$MySqlDatabaseName Database droped<br />";
}
```

```
else
{
     echo "Error creating database: " . mysql_error() . "<br />";
     die;
}

// Create  database
if (mysql_query("CREATE DATABASE $MySqlDatabaseName",$con))
{
    echo "$MySqlDatabaseName Database created<br />";
}
else
{
     echo "Error creating database: " . mysql_error() . "<br />";
     die;
}

// Select database
mysql_select_db("$MySqlDatabaseName", $con);


//Output a line of the file until the end is reached
$file = fopen("$inputCSVFilename", "r");
for ($row = 1;($csvData = fgetcsv($file, $delimiter= ",")) !== FALSE; $row++)
{
    $num = count($csvData);
//    echo "<p> $num fields in line $row: <br /></p>\n";

    for ($field=0; $field < $num; $field++)
    {
//        echo "row $row field $field ";
//        echo $csvData[$field] . "<br />\n";
        if( ($num != $expectedNumberOfFields))
        {
            echo "Expected $expectedNumberOfFields read $num!<br />";
            die;
        }
        if( ($num == $expectedNumberOfFields) && ($row == 1))
```

```
        {
            $csvField[$field] = $csvData[$field];
            if($field == 0)
            {
                // Create table in  database
                $sql = "CREATE TABLE $MySqlTableName
                (
                    tableID int,
                    $csvData[$field] $MySqlDatabaseSizes[$field]
                )";
                $error = mysql_query($sql,$con);
                if(strlen($error) > 1)
                    echo "Error: " . mysql_error() . "<br />";
            }
            else
            {
                // add next column table in  database
                $sql = "ALTER TABLE $MySqlTableName ADD $csvData[$field] $MySqlDatabaseSizes[$field]
NULL";
                $error = mysql_query($sql,$con);
                if(strlen($error) > 1)
                    echo "Error: " . mysql_error() . "<br />";
            }
        }
        if( ($num == $expectedNumberOfFields) && ($row > 1))
        {
            if($field == 0)
            {
//                echo "insert $csvField[$field]: $csvData[$field]<br />";
                $sql = "INSERT INTO $MySqlTableName (tableID, $csvField[$field]) VALUES ($row - 1,
'$csvData[$field]')";
//                echo "$sql<br />";
                $error = mysql_query($sql,$con);
                if(strlen($error) > 1)
                    echo "Error: " . mysql_error() . "<br />";
            }
            else
            {
```

```
//                echo "update $csvField[$field]: $csvData[$field]<br />";
                $sql = "UPDATE $MySqlTableName SET $csvField[$field] = $csvData[$field] WHERE tableID
= $row - 1";
//                echo "$sql<br />";
                $error = mysql_query($sql,$con);
                if(strlen($error) > 1)
                    echo "Error: " . mysql_error() . "<br />";
            }
        }
    }


}
fclose($file);
mysql_close($con);


// Querying the MySQL Database so the select boxes can be populated
$MySqlServerName = "localhost";
$MySqlDatabaseName = "tjm_db";
$MySqlTableName = "eda_table";

// Database stuff
// connect to MySql server
$con = mysql_connect("$MySqlServerName","root","admin");
if (!$con)
{
    die('Could not connect: ' . mysql_error()) . "<br />";
}
else
{
    echo "Connected to MySql Server $MySqlServerName<br />";
}

// Select database
mysql_select_db("$MySqlDatabaseName", $con);

echo "<form action=\"tjm_form_action.php\" method=\"post\">";
```

```
echo "<table border=\"0\" width=\"$tableWidth\" cellspacing=\"0\" cellpadding=\"3\">";
echo "      <tr>";
echo "            <td colspan=\"$numColumnsInForm\" bgcolor=\"$bgColor\" width=\"$tableWidth\">";
echo "            <p align=\"center\"><b>User-Defined Exploratory Data Analysis</b></td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td colspan=\"$numColumnsInForm\" bgcolor=\"$bgColor\" width=\"$tableWidth\">";
echo "            <b class=\"timh1\"><font size=\"2\">This form allows users to explore ";
echo "dimensions of their GIS metadata.  Data collected transcend various data ";
echo "scales (nominal and ratio).  Users can make compound queries of the data.  Select and
populate all of the appropriate fields that you wish to ";
echo "query.  Leaving all fields blank will return all records. </font> </b></td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td colspan=\"$numColumnsInForm\" bgcolor=\"$bgColor\"
width=\"$tableWidth\"> </td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td colspan=\"$numColumnsInForm\" bgcolor=\"$bgColorRequired\"
width=\"<$tableWidth\">";
echo "            <p align=\"center\"><b>Required Elements</b></td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><i>Data Set
Title:</i></td>";
//echo "          <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"dataSetTitle\" ></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"dataSetTitle\" >";

// Autopopulate the Data Set Title Drop Down Box
$sql = "SELECT DISTINCT r01_Data_Set_Title from $MySqlTableName ORDER by r01_Data_Set_Title ASC";
$result= mysql_query($sql,$con);

      echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{
```

```
      echo "<option value= '" . $row['r01_Data_Set_Title'] . "' >" . $row['r01_Data_Set_Title'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "            </td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>Search
Pattern</td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"dataSetTitlePattern\" >*</td>";
//echo "          <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"></td>";
echo "      <tr>";

// Publication Date Stuff
echo "            <td width=\"$tableWidthColumn1Required\"
bgcolor=\"$bgColorRequired\"><i>Publication Date:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> After:  ";
echo "            <input type = text name = \"pubDateLower\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> Before:  ";
echo "            <input type = text name = \"pubDateUpper\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"> NOT FOUND
<input type = 'checkbox' name = 'pubDateNF' value = 'checked'></td>";


// Language Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Language:</i></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"languageQuery\" ></td>";
echo "      <td bgcolor=\"$bgColorRequired\"><td bgcolor=\"$bgColorRequired\"></tr>";

// Data Theme Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>Data
Theme:</i></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"themeQuery\" >*</td>";
```

```
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'datathemeNF' value =
'checked'></td></tr>";

// Abtract Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Abstract:</td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"abstractQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'abstractNF' value =
'checked'></td></tr>";

// Metadata POC Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata POC:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"metadataPOC\" >";

// Autopopulate the Metadata POC Drop Down Box
$sql = "SELECT DISTINCT r06_Metadata_POC from $MySqlTableName ORDER by r06_Metadata_POC ASC";
$result= mysql_query($sql,$con);

echo "<option value = 'NONE'>NONE";


while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['r06_Metadata_POC'] . "' >" . $row['r06_Metadata_POC'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "            </td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";
```

```
// Metadata Date Stuff
echo "            <td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><i>Metadata
Date:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> After:  ";
echo "            <input type = text name = \"mdDateLower\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> Before:  ";
echo "            <input type = text name = \"mdDateUpper\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"> NOT FOUND
<input type = 'checkbox' name = 'mdDateNF' value = 'checked'></td>";

$bgColorRequired = "yellow";
echo "       <tr><td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><td
width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><td
width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\">";

//  FGDC Suggested Themes Label
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"></i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><i><b>FGDC-
Suggested Elements:</b></i></td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


// Spatial Resolution Stuff
echo "            <tr><td width=\"$tableWidthColumn1Required\"
bgcolor=\"$bgColorRequired\"><i>Spatial Resolution:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> Greater
Than:  ";
echo "            <input type = text name = \"spatialLowerQuery\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> Less Than:
";
echo "            <input type = text name = \"spatialUpperQuery\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"> NOT FOUND
<input type = 'checkbox' name = 'spatialNF' value = 'checked'></td>";
```

```
//  Distribution Format Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Distribution Format:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"distributionFormat\" >";

// Autopopulate the Distribution Format Drop Down Box
$sql = "SELECT DISTINCT s01_Distribution_Format from $MySqlTableName ORDER by s01_Distribution_Format
ASC";
$result= mysql_query($sql,$con);
      echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['s01_Distribution_Format'] . "' >" .
$row['s01_Distribution_Format'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


// Additional Spatial Information Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Additional Spatial Information:</i></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"additionalSpatialQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'additionalspatialNF' value =
'checked'></td></tr>";


//  Spatial Representation Stuff
```

```
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Spatial Representation:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"representationQuery\" >";

// Autopopulate the Spatial Representation Drop Down Box
$sql = "SELECT DISTINCT s03_Spatial_Representation from $MySqlTableName ORDER by
s03_Spatial_Representation ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['s03_Spatial_Representation'] . "' >" .
$row['s03_Spatial_Representation'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Reference Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Reference System:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"referenceQuery\" >";

// Autopopulate the Spatial Representation Drop Down Box
$sql = "SELECT DISTINCT s04_Reference_System from $MySqlTableName ORDER by s04_Reference_System ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{
```

```
      echo "<option value= '" . $row['s04_Reference_System'] . "' >" . $row['s04_Reference_System'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

// Lineage Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Lineage Statement:</td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"lineageQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'lineageNF' value =
'checked'></td></tr>";

// Online Resource Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>Online
Resource:</td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"onlineQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'onlineNF' value =
'checked'></td></tr>";

// Metadata Field Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Field:</td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"mdfieldQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'mdfieldNF' value =
'checked'></td></tr>";
```

```
//  Metadata Standard Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Standard:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"mdstandardQuery\" >";

// Autopopulate the Spatial Representation Drop Down Box
$sql = "SELECT DISTINCT s08_Metadata_Standard_Name from $MySqlTableName ORDER by
s08_Metadata_Standard_Name ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['s08_Metadata_Standard_Name'] . "' >" .
$row['s08_Metadata_Standard_Name'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


//  Metadata Version Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Version:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"mdversionQuery\" >";

// Autopopulate the Metadata Version Drop Down Box
$sql = "SELECT DISTINCT s09_Metadata_Standard_Version from $MySqlTableName ORDER by
s09_Metadata_Standard_Version ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";
```

```
while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['s09_Metadata_Standard_Version'] . "' >" .
$row['s09_Metadata_Standard_Version'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

// Metadata Language Stuff
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Language:</i></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"languageQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\"><td bgcolor=\"$bgColorRequired\"></tr>";


//  Metadata Character
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Character:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"mdcharacterQuery\" >";

// Autopopulate the Metadata Character Set Drop Down Box
$sql = "SELECT DISTINCT s11_Metadata_Character_Set from $MySqlTableName ORDER by
s11_Metadata_Character_Set ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{
```

```
      echo "<option value= '" . $row['s11_Metadata_Character_Set'] . "' >" .
$row['s11_Metadata_Character_Set'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


//  Location of Data
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Location of Data:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> Latitude
(+/- DD):   ";
echo "            <input type = text name = \"locationLatitude\" size=\"12\" >";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\">Longitude
(+/- DD):  ";
echo "            <input type = text name = \"locationLongitude\" size=\"12\" ></td><td
width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><td
width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"></tr>";

echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><td
width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> BUFFER:  ";
echo "            <input type = text name = \"locationBuffer\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"locationDistance\" >";
echo "      <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"> NOT FOUND <input
type = 'checkbox' name = 'locationNF' value = 'checked'></td></tr>";



//  Responsible Party
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Responsible Party:</i></td>";
```

```
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"partyQuery\" >";

// Autopopulate the Responsible Party Drop Down Box
$sql = "SELECT DISTINCT s14_Responsible_Party from $MySqlTableName ORDER by s14_Responsible_Party
ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['s14_Responsible_Party'] . "' >" . $row['s14_Responsible_Party']
;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


//  Character Set
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Character Set:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"characterQuery\" >";

// Autopopulate the Character Set Drop Down Box
$sql = "SELECT DISTINCT s15_Data_Set_Character_Set from $MySqlTableName ORDER by
s15_Data_Set_Character_Set ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{
```

```
      echo "<option value= '" . $row['s15_Data_Set_Character_Set'] . "' >" .
$row['s15_Data_Set_Character_Set'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


$bgColorRequired = "beige";
echo "       <tr><td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><td
width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><td
width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\">";

//  Non-Required Label
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"></i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><i><b>Non
Required Elements:</b></i></td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


// Area of Layer
echo "            <tr><td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><i>Area
of Layer:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> Greater
Than:  ";
echo "            <input type = text name = \"areaLower\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> Less Than:
";
echo "            <input type = text name = \"areaUpper\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"areaNF\" >";
```

```
// Place Query
echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>Place
Query:</i></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"placeQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\"><td bgcolor=\"$bgColorRequired\"></tr>";


//  Update Frequency
echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>Update
Frequency:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"updateQuery\" >";

// Autopopulate the Update Frequency Drop Down Box
$sql = "SELECT DISTINCT n03_updatefrequency from $MySqlTableName ORDER by n03_updatefrequency ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n03_updatefrequency'] . "' >" . $row['n03_updatefrequency'] ;

}
echo "</SELECT>";
// End of autopopulating the drop down box

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Geoid
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Geoid:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"geoidQuery\" >";
```

```
// Autopopulate the Geoid Drop Down Box
$sql = "SELECT DISTINCT n05_geoid from $MySqlTableName ORDER by n05_geoid ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n05_geoid'] . "' >" . $row['n05_geoid'] ;

}
echo "</SELECT>";


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Ellipsoid
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Ellipsoid:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"ellipsoidQuery\" >";

// Autopopulate the Ellipsoid Drop Down Box
$sql = "SELECT DISTINCT n06_ellipsoid from $MySqlTableName ORDER by n06_ellipsoid ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n06_ellipsoid'] . "' >" . $row['n06_ellipsoid'] ;

}
echo "</SELECT>";
```

```
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Spatial Data Organization
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Spatial Data Organization:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"sdorganizationQuery\" >";

// Autopopulate the Spatial Data Organization Drop Down Box
$sql = "SELECT DISTINCT n09_sdorganization from $MySqlTableName ORDER by n09_sdorganization ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n09_sdorganization'] . "' >" . $row['n09_sdorganization'] ;

}
echo "</SELECT>";


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  SDTS Type
echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>SDTS
Type:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"sdtsQuery\" >";

// Autopopulate the Spatial Data Organization Drop Down Box
$sql = "SELECT DISTINCT n10_sdtstype from $MySqlTableName ORDER by n10_sdtstype ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
```

```
{

      echo "<option value= '" . $row['n10_sdtstype'] . "' >" . $row['n10_sdtstype'] ;

}
echo "</SELECT>";

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";


// Number of Features Stuff
echo "            <tr><td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><i>Number
of Features:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> Greater
Than:  ";
echo "            <input type = text name = \"numfeaturesLower\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> Less Than:
";
echo "            <input type = text name = \"numfeaturesUpper\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"> NOT FOUND
<input type = 'checkbox' name = 'numfeaturesNF' value = 'checked'></td>";


//  Attribute Definition System
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Attribute Definition:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"attributedefinitionQuery\" >";

// Autopopulate the Attribute Definition Drop Down Box
$sql = "SELECT DISTINCT n12_attdefsystem from $MySqlTableName ORDER by n12_attdefsystem ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{
```

```
      echo "<option value= '" . $row['n12_attdefsystem'] . "' >" . $row['n12_attdefsystem'] ;

}
echo "</SELECT>";

echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Contact Organization
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Contact Organization:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"contactorgQuery\" >";

// Autopopulate the Contact Organization Drop Down Box
$sql = "SELECT DISTINCT n13_contactorganization from $MySqlTableName ORDER by n13_contactorganization
ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n13_contactorganization'] . "' >" .
$row['n13_contactorganization'] ;

}
echo "</SELECT>";


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Metadata Organization
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Organizatoin:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"metadataorgQuery\" >";
```

```
// Autopopulate the Metadata Organization Drop Down Box
$sql = "SELECT DISTINCT n14_mdorganization from $MySqlTableName ORDER by n14_mdorganization ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n14_mdorganization'] . "' >" . $row['n14_mdorganization'] ;

}
echo "</SELECT>";


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Contact Position
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Contact Position:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"contactposQuery\" >";

// Autopopulate the Contact Position Drop Down Box
$sql = "SELECT DISTINCT n15_contactposition from $MySqlTableName ORDER by n15_contactposition ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n15_contactposition'] . "' >" . $row['n15_contactposition'] ;

}
echo "</SELECT>";
```

```
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

//  Metadata Position
echo "            <tr><td width=\"$tableWidthColumn3Required\"
bgcolor=\"$bgColorRequired\"><i>Metadata Position:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"><SELECT  name
= \"metadataposQuery\" >";

// Autopopulate the Metadata Position Drop Down Box
$sql = "SELECT DISTINCT n16_mdposition from $MySqlTableName ORDER by n16_mdposition ASC";
$result= mysql_query($sql,$con);
            echo "<option value = 'NONE'>NONE";

while($row = mysql_fetch_array($result))
{

      echo "<option value= '" . $row['n16_mdposition'] . "' >" . $row['n16_mdposition'] ;

}
echo "</SELECT>";


echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"></td></tr>";

// Use Constraints
echo "            <tr><td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"><i>Use
Constraints:</td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"><input type =
text size = 30 name = \"useQuery\" >*</td>";
echo "      <td bgcolor=\"$bgColorRequired\">   <td width=\"$tableWidthColumn4Required\"
bgcolor=\"$bgColorRequired\"> NOT FOUND <input type = 'checkbox' name = 'useNF' value =
'checked'></td></tr>";


// Number of Attributes Stuff
```

```
echo "            <tr><td width=\"$tableWidthColumn1Required\" bgcolor=\"$bgColorRequired\"><i>Number
of Attributes:</i></td>";
echo "            <td width=\"$tableWidthColumn2Required\" bgcolor=\"$bgColorRequired\"> Greater
Than:  ";
echo "            <input type = text name = \"numattributesLower\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn3Required\" bgcolor=\"$bgColorRequired\"> Less Than:
";
echo "            <input type = text name = \"numattributesUpper\" size=\"12\" ></td>";
echo "            <td width=\"$tableWidthColumn4Required\" bgcolor=\"$bgColorRequired\"> NOT FOUND
<input type = 'checkbox' name = 'numattributesNF' value = 'checked'></td>";



echo "      </tr>";
echo "      <tr>";
echo "            <td width=\"$tableWidth\" colspan=\"$numColumnsInForm\"
bgcolor=\"$bgColorRequired\"> </td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td width=\"$tableWidth\" colspan=\"$numColumnsInForm\"
bgcolor=\"$bgColorRequired\">";
echo "            <p align=\"center\"><font size=\"1\">* Uses keyword matching to return applicable
";
echo "            records</font></td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td width=\"$tableWidth\" colspan=\"$numColumnsInForm\">";
echo "             </td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td width=\"$tableWidth\" colspan=\"$numColumnsInForm\">";
echo "                  <p align=\"center\"><input type=\"submit\" value=\"Submit\"
name=\"B1\"><input type=\"reset\" value=\"Reset\" name=\"B2\"></td>";
echo "      </tr>";
echo "      <tr>";
echo "            <td width=\"$tableWidth\" colspan=\"$numColumnsInForm\">";
echo "                   </td>";
echo "      </tr>";
```

```
echo "</form>";
?>

</body>
</html>
```

APPENDIX F: PHP CODE USED TO QUERY MYSQL DATABASE BASED ON USER PARAMETERS FROM HTML FORM ELEMENTS AND DISPLAY RESULTS IN WEB PAGE

```
<html>
<body>
<?php

$MySqlServerName = "localhost";
$MySqlDatabaseName = "tjm_db";
$MySqlTableName = "eda_table";

echo "dataSetTitle: " . $_POST["dataSetTitle"] . "<br />";
echo "dataSetTitlePattern: " . $_POST["dataSetTitlePattern"] . "<br />";
echo "pubDateLower: " . $_POST["pubDateLower"] . "<br />";
echo "pubDateUpper: " . $_POST["pubDateUpper"] . "<br />";
echo "pubDateNotFound: " . $_POST["pubDateNF"] . "<br />";
echo "spatialLower: " . $_POST["spatialLower"] . "<br />";
echo "spatialUpper: " . $_POST["spatialUpper"] . "<br />";

$dateLower = $_POST["pubDateLower"];
$dateUpper = $_POST["pubDateUpper"];
$dataSetTitle = $_POST["dataSetTitle"];
$pattern = $_POST["dataSetTitlePattern"];
$title = $_POST["dataSetTitle"];
$pubDateNotFound = $_POST["pubDateNF"];
$dataTheme = $_POST["themeQuery"];
$dataThemeNF = $_POST["datathemeNF"];
$abstract = $_POST["abstractQuery"];
$abstractNF = $_POST["abstractNF"];
$mdDateLower = $_POST["mdDateLower"];
$mdDateUpper = $_POST["mdDateUpper"];
$mdDateNF = $_POST["mdDateNF"];
$spatialLower = $_POST["spatialLowerQuery"];
$spatialUpper = $_POST["spatialUpperQuery"];
$spatialNF = $_POST["spatialNF"];
```

```
$additionalSpatial = $_POST["additionalSpatialQuery"];
$additionalSpatialNF = $_POST["additionalspatialNF"];
$lineageStatement = $_POST["lineageQuery"];
$lineageNF = $_POST["lineageNF"];
$onlineQuery = $_POST["onlineQuery"];
$onlineNF = $_POST["onlineNF"];
$mdfieldQuery = $_POST["mdfieldQuery"];
$mdfieldNF = $_POST["mdfieldNF"];
$areaUpper = $_POST["areaUpper"];
$areaLower = $_POST["areaLower"];
$areaNF = $_POST["areaNF"];
$placeQuery = $_POST["placeQuery"];
$numfeaturesLower = $_POST["numfeaturesLower"];
$numfeaturesUpper = $_POST["numfeaturesUpper"];
$numfeaturesNF = $_POST["numfeaturesNF"];
$useQuery = $_POST["useQuery"];
$useNF = $_POST["useNF"];
$numattributesLower = $_POST["numattributesLower"];
$numattributesUpper = $_POST["numattributesUpper"];
$numattributesNF = $_POST["numattributesNF"];
$languageQuery = $_POST["languageQuery"];

// Stuff from drop down boxes
$metadataPOC = $_POST["metadataPOC"];
$distributionFormat = $_POST["distributionFormat"];
$spatialRepresentation = $_POST["representationQuery"];
$referenceSystem = $_POST["referenceQuery"];
$metadataStandard = $_POST["mdstandardQuery"];
$metadataPOC = $_POST["metadataPOC"];
$metadataVersion =  $_POST["mdversionQuery"];
$metadataLanguage =  $_POST["languageQuery"];
$metadataCharacter =  $_POST["mdcharacterQuery"];
$responsibleParty = $_POST["partyQuery"];
$characterSet = $_POST["characterQuery"];

// connect to MySql server
$con = mysql_connect("$MySqlServerName","root","admin");
if (!$con)
```

```
{
    die('Could not connect: ' . mysql_error()) . "<br />";
}
else
{
    echo "Connected to MySql Server $MySqlServerName<br />";
}

// Select database
mysql_select_db("$MySqlDatabaseName", $con);

$sql = "SELECT * from $MySqlTableName WHERE tableID > 0";
echo $dataSetTitle;


if($dataSetTitle != 'NONE')
{
      $sql .= " AND r01_Data_Set_Title = '$dataSetTitle'";
}

if(strlen($pattern) > 0)
{
      $sql .= " AND r01_Data_Set_Title LIKE '%$pattern%'";
}


if(strlen($dateLower) > 0)
{
      $sql .= " AND r02_Publication_Date > $dateLower";
}


if(strlen($dateUpper) > 0)
{
      $sql .= " AND r02_Publication_Date < $dateUpper";
}
```

```
if($pubDateNotFound)
{
      $sql .= " AND r02_Publication_Date = 'NOT FOUND' ";
}


if(strlen($languageQuery) > 0)
{
      $sql .= " AND r03_Language LIKE '%$languageQuery%'";
}


if(strlen($dataTheme) > 0)
{
      $sql .= " AND r04_Data_Theme LIKE '%$dataTheme%'";
}


if($dataThemeNF)
{
      $sql .= " AND r04_Data_Theme = 'NOT FOUND' ";
}

if(strlen($abstract) > 0)
{
      $sql .= " AND r05_Abstract LIKE '%$abstract%'";
}


if($abstractNF)
{
      $sql .= " AND r05_Abstract = 'NOT FOUND' ";
}


if($metadataPOC != 'NONE')
{
      $sql .= " AND r06_Metadata_POC = '$metadataPOC'";
```

```
}

if(strlen($mdDateLower) > 0)
{
      $sql .= " AND r07_Metadata_Date > $mdDateLower";
}


if(strlen($mdDateUpper) > 0)
{
      $sql .= " AND r07_Metadata_Date < $mdDateUpper";
}


if($mdDateNF)
{
      $sql .= " AND r07_Metadata_Date = 'NOT FOUND' ";
}


//does not seem to like me
if(strlen($spatialLower) > 0)
{
      $sql .= " AND s00_Spatial_Resolution > $spatialLower";
}

//does not seem to like me
if(strlen($spatialUpper) > 0)
{
      $sql .= " AND s00_Spatial_Resolution < $spatialUpper";
}


if($spatialNF)
{
      $sql .= " AND s00_Spatial_Resolution = 'unknown' ";
}
```

```
if($distributionFormat != 'NONE')
{
      $sql .= " AND s01_Distribution_Format = '$distributionFormat'";
}



if(strlen($additionalSpatial) > 0)
{
      $sql .= " AND s02_Additional_Spatial LIKE '%$additionalSpatial%'";
}


if($additionalSpatialNF)
{
      $sql .= " AND s02_Additional_Spatial = 'NOT FOUND' ";
}

if($spatialRepresentation != 'NONE')
{
      $sql .= " AND s03_Spatial_Representation = '$spatialRepresentation'";
}

if($referenceSystem != 'NONE')
{
      $sql .= " AND s04_Reference_System = '$referenceSystem'";
}


if(strlen($lineageStatement) > 0)
{
      $sql .= " AND s05_Lineage_Statement LIKE '%$lineageStatement%'";
}


if($lineageNF)
{
```

```
	$sql .= " AND s05_Lineage_Statement = 'NOT FOUND' ";
}


if(strlen($onlineQuery) > 0)
{
	$sql .= " AND s06_Online_Resource LIKE '%$onlineQuery%'";
}


if($onlineNF)
{
	$sql .= " AND s06_Online_Resource = 'NOT FOUND' ";
}


if(strlen($mdfieldQuery) > 0)
{
	$sql .= " AND s07_Metadata_Field LIKE '%$mdfieldQuery%'";
}


if($mdfieldNF)
{
	$sql .= " AND s07_Metadata_Field = 'NOT FOUND' ";
}


if($metadataStandard != 'NONE')
{
	$sql .= " AND s08_Metadata_Standard_Name = '$metadataStandard'";
}

if($metadataVersion != 'NONE')
{
	$sql .= " AND s09_Metadata_Standard_Version = '$metadataVersion'";
}
```

```
if(strlen($metadataLanguage) > 0)
{
      $sql .= " AND s10_Metadata_Language LIKE '%$metadataLanguage%'";
}


if($metadataCharacter != 'NONE')
{
      $sql .= " AND s11_Metadata_Character_Set = '$metadataCharacter'";
}

if($responsibleParty != 'NONE')
{
      $sql .= " AND s14_Responsible_Party = '$responsibleParty'";
}


if($characterSet != 'NONE')
{
      $sql .= " AND s15_Data_Set_Character_Set = '$characterSet'";
}

// Beginning of non-required query stuff
$updateQuery = $_POST["updateQuery"];
$geoidQuery = $_POST["geoidQuery"];
$ellipsoidQuery = $_POST["ellipsoidQuery"];
$sdorganizationQuery = $_POST["sdorganizationQuery"];
$sdtsQuery = $_POST["sdtsQuery"];
$attributedefinitionQuery = $_POST["attributedefinitionQuery"];
$contactorgQuery = $_POST["contactorgQuery"];
$metadataorgQuery = $_POST["metadataorgQuery"];
$contactposQuery = $_POST["contactposQuery"];
$metadataposQuery = $_POST["metadataposQuery"];


if(strlen($areaLower) > 0)
{
```

```
	$sql .= " AND n01_areaofextentsqmi > $areaLower";
}

if(strlen($areaUpper) > 0)
{
	$sql .= " AND n01_areaofextentsqmi < $areaUpper";
}

if($areaNF)
{
	$sql .= " AND n01_areaofextentsqmi = 'unknown' ";
}

if(strlen($placeQuery) > 0)
{
	$sql .= " AND n04_placekey LIKE '%$placeQuery%'";
}


if($updateQuery != 'NONE')
{
	$sql .= " AND n03_updatefrequency = '$updateQuery'";
}


if($geoidQuery  != 'NONE')
{
	$sql .= " AND n05_geoid = '$geoidQuery '";
}


if($ellipsoidQuery != 'NONE')
{
	$sql .= " AND n06_ellipsoid = '$ellipsoidQuery'";
}


if($sdorganizationQuery != 'NONE')
```

```
{
      $sql .= " AND n09_sdorganization = '$sdorganizationQuery'";
}


if($sdtsQuery != 'NONE')
{
      $sql .= " AND n10_sdtstype = '$sdtsQuery'";
}



if(strlen($numfeaturesLower) > 0)
{
      $sql .= " AND n11_objectcount > $numfeaturesLower";
}

if(strlen($numfeaturesUpper) > 0)
{
      $sql .= " AND n11_objectcount < $numfeaturesUpper";
}

if($numfeaturesNF)
{
      $sql .= " AND n11_objectcount LIKE '%HASH%' ";
}

if($attributedefinitionQuery != 'NONE')
{
      $sql .= " AND n12_attdefsystem = '$attributedefinitionQuery'";
}

if($contactorgQuery != 'NONE')
{
      $sql .= " AND n13_contactorganization = '$contactorgQuery'";
}

if($metadataorgQuery != 'NONE')
```

```
{
      $sql .= " AND n14_mdorganization = '$metadataorgQuery'";
}

if($contactposQuery != 'NONE')
{
      $sql .= " AND n15_contactposition = '$contactposQuery'";
}

if($metadataposQuery != 'NONE')
{
      $sql .= " AND n16_mdposition = '$metadataposQuery'";
}

if(strlen($useQuery) > 0)
{
      $sql .= " AND n21_useconstraints LIKE '%$useQuery%'";
}

if($useNF)
{
      $sql .= " AND n21_useconstraints = 'NOT FOUND' ";
}

if(strlen($numattributesLower) > 0)
{
      $sql .= " AND n23_numattributes > $numattributesLower";
}

if(strlen($numattributesUpper) > 0)
{
      $sql .= " AND n23_numattributes < $numattributesUpper";
}

if($numattributesNF)
{
      $sql .= " AND n23_numattributes = 'unknown' ";
}
```

```
echo "SQL: $sql<br>";

$result= mysql_query($sql,$con);
//$result = mysql_query("SELECT * FROM $MySqlTableName");
//if(strlen($result) > 1)
//    echo "Error: " . mysql_error() . "<br />";




$countFound = 0;

echo "<table border = 2>";
 echo "<tr><td><center><b>File Name<td><center><b>Data Set Title<td><center><b>Publication
Date<td><center><b>Language<td><center><b>Data Theme<td><center><b>Abstract<td><center><b>Metadata
POC<td><center><b>Metadata Date<td><center><b>Spatial Resolution<td><center><b>Distribution
Format<td><center><b>Additional Spatial <BR> Information<td><center><b>Spatial <BR>
Representation<td><center><b>Reference <BR> System<td><center><b>Lineage<td><center><b>Online
Resource<td><center><b>Metadata Field<td><center><b>Metadata Standard<td><center><b>Metadata
Version<td><center><b>Metadata Language<td><center><b>Metadata Character<td><center><B>Location <BR>
of Data<td><center><b>Responsible Party<td><center><b>Character Set<td><center><b>Area of <BR>
Layer";
echo "<td><center><b>Place Key<td><center><b>Update <BR>
Frequency<td><center><b>Geoid<td><center><b>Ellipsoid<td><center><b>Spatial Data <BR>
Organization<td><center><b>SDTS Type<td><center><b>Number of <BR> Features<td><center><b>Attribute
Definition<td><center><b>Contact <BR> Organization<td><center><b>Metadata <BR>
Organization<td><center><b>Contact <BR> Position<td><center><b>Metadata <BR>
Position<td><center><b>Use Constraints<td><center><b>Number of Attributes</tr>";

while($row = mysql_fetch_array($result))
{
```

```
      echo "<tr><td>" . $row['File_Name'] . "<TD bgcolor = '#CCFFCC'>" . $row['r01_Data_Set_Title'] .
"<TD bgcolor = '#CCFFCC'>" . $row['r02_Publication_Date'] . "<td bgcolor = '#CCFFCC'>" .
$row['r03_Language'] . "<td bgcolor = '#CCFFCC'>" . $row['r04_Data_Theme'] . "<td bgcolor =
'#CCFFCC'>" . $row['r05_Abstract'] . "<td bgcolor = '#CCFFCC'>" . $row['r06_Metadata_POC'] . "<td
bgcolor = '#CCFFCC'>" . $row['r07_Metadata_Date'] . "<td bgcolor = '#FFFF00'>" .
$row['s00_Spatial_Resolution'] . "<td bgcolor = '#FFFF00'>" . $row['s01_Distribution_Format'] ."<td
bgcolor = '#FFFF00'>" . $row['s02_Additional_Spatial'] . "<td bgcolor = '#FFFF00'>" .
$row['s03_Spatial_Representation'] . "<td bgcolor = '#FFFF00'>" . $row['s04_Reference_System'] . "<td
bgcolor = '#FFFF00'>" . $row['s05_Lineage_Statement'] . "<td bgcolor = '#FFFF00'>" .
$row['s06_Online_Resource'] . "<td bgcolor = '#FFFF00'>" . $row['s07_Metadata_Field'] . "<td bgcolor
= '#FFFF00'>" . $row['s08_Metadata_Standard_Name'] . "<td bgcolor = '#FFFF00'>" .
$row['s09_Metadata_Standard_Version'] . "<td bgcolor = '#FFFF00'>" . $row['s10_Metadata_Language'] .
"<td bgcolor = '#FFFF00'>" . $row['s11_Metadata_Character_Set'] . "<td bgcolor = '#FFFF00'>" .
$row['s12_locationlong'] . ", " . $row['s13_locationlat'] . "<td bgcolor = '#FFFF00'>" .
$row['s14_Responsible_Party'] . "<td bgcolor = '#FFFF00'>" . $row['s15_Data_Set_Character_Set'] .
"<td bgcolor = '#E4E0AD'>" . $row['n01_areaofextentsqmi'] . "<td bgcolor = '#E4E0AD'>" .
$row['n04_placekey'] . "<td bgcolor = '#E4E0AD'>" . $row['n03_updatefrequency'] . "<td bgcolor =
'#E4E0AD'>" . $row['n05_geoid'] . "<td bgcolor = '#E4E0AD'>" . $row['n06_ellipsoid'] . "<td bgcolor =
'#E4E0AD'>" . $row['n09_sdorganization'] . "<td bgcolor = '#E4E0AD'>" . $row['n10_sdtstype'] . "<td
bgcolor = '#E4E0AD'>" . $row['n11_objectcount'] . "<td bgcolor = '#E4E0AD'>" .
$row['n12_attdefsystem'] ;
echo "<td bgcolor = '#E4E0AD'>" . $row['n13_contactorganization'] . "<td bgcolor = '#E4E0AD'>" .
$row['n14_mdorganization'] . "<td bgcolor = '#E4E0AD'>" . $row['n15_contactposition'] . "<td bgcolor
= '#E4E0AD'>" . $row['n16_mdposition'] . "<td bgcolor = '#E4E0AD'>" . $row['n21_useconstraints'] .
"<td bgcolor = '#E4E0AD'>" . $row['n23_numattributes'] ."</tr>";
      $countFound += 1;

}

echo "<tr><td>END</td></tr>";

echo "</table><br>";
echo $countFound . " records were returned from this query.";

 mysql_close($con);

?>
```

```
</body>
</html>
```